POZNAN UNIVERSITY OF TECHNOLOGY

Data-efficient and explainable machine learning in visual perception for autonomous vehicles

by

Tomasz Nowak

in the

Institute of Robotics and Machine Intelligence

Faculty of Control, Robotics and Electrical Engineering

Supervisor: Prof. Piotr Skrzypczyński, Ph.D., D.Sc.

Auxiliary Supervisor: Michał Nowicki, Ph.D., D.Sc.

November 3, 2024

Abstract

The development of data-efficient and explainable machine learning models is important for the enhancement of the safety and reliability of autonomous driving systems. Visual perception, a core component of autonomous vehicles, relies heavily on deep neural networks to interpret complex and dynamic environments. However, traditional deep neural networks often require extensive labeled datasets to achieve high accuracy, posing challenges in scenarios where data collection is limited or expensive. Additionally, the inherent opacity of these models hinders comprehension of their decision-making processes, raising concerns about safety and trust. This dissertation addresses these challenges by proposing methods that enhance data efficiency and explainability within the context of visual perception for autonomous vehicles.

This research introduces several contributions aimed at improving the performance and interpretability of deep neural networks in autonomous driving applications. One of the primary contributions is a guided learning framework that enhances model training by using targeted feedback from visualizations of the networks attention. Moreover, novel neural network architectures was developed for the estimation of geometric features from monocular images. These architectures incorporate in its loss functions 3D object models and environmental geometric constraints to enhance the accuracy of keypoint localization to make predictions that are consistent with the physical structure of the objects being observed and providing a robust solution even in scenarios with limited training data.

Furthermore, the dissertation investigates the integration of uncertainty estimation within neural network architectures. A dedicated neural network was designed to predict the uncertainty of 2D and 3D keypoint coordinates, offering a probabilistic measure of confidence in each prediction. Additionally, the dissertation presents a processing pipeline for estimating the pose of surrounding vehicles using monocular images. This pipeline integrates the Unscented Transform algorithm to propagate uncertainties from 2D and 3D keypoint estimations, providing a measure of the overall uncertainty in vehicle pose estimations.

The proposed methods were evaluated in two real-world scenarios. The first scenario involved a docking maneuver to a charging station using an electric city bus. In this application, the system successfully estimate the pose of the bus relative to the charging station, enabling precise docking with an error margin of less than 30 cm from a distance of over 30 meters. The second scenario focuses on the pose estimation of surrounding vehicles in a city environment. The used dataset is an ApolloCar3D that provides images of an urban environment to test the pose estimation capabilities of the developed models. The proposed method achieved state-of-the-art results on the aforementioned dataset.

The methods and their experimental evaluation presented in this dissertation validates research theses: Firstly, architectures that extract and visualise meaningful intermediate features can enhance learning by augmenting existing datasets. Secondly, they accurately describe the uncertainty of the produced geometric features. Thirdly, utilising available 3D models of observed objects facilitates the learning of geometric features from 2D images without exact labeling, and significantly improves the detection accuracy of these features. Lastly, knowledge of geometric constraints from known object models helps reduce false feature detection and increases the precision of feature localization.

Streszczenie

Rozwój modeli uczenia maszynowego, które są i wydajne w kontekście danych uczących i łatwe w interpretacji, jest ważny dla zwiększenia bezpieczeństwa i niezawodności autonomicznych pojazdów. Moduły percepcji wizualnej w pojazdach autonomicznych w celu interpretacji złożonych i dynamicznych środowisk wykorzystują w dużej mierze głębokie sieci neuronowe, które często wymagają obszernych zbiorów danych, aby osiągnąć wysoką dokładność. Stanowi to wyzwanie w scenariuszach, w których gromadzenie danych jest problematyczne lub kosztowne. Dodatkowo, brak transparentności tych modeli utrudnia zrozumienie ich procesów decyzyjnych, budząc obawy o bezpieczeństwo i wiarygodność. Niniejsza rozprawa doktorska odnosi się do tych wyzwań, proponując metody, które zwiększają efektywność wykorzystania danych uczących i wyjaśnialność decyzji w kontekście percepcji wizualnej dla pojazdów autonomicznych.

Badania te wprowadzają kilka rozwiązań mających na celu poprawę wydajności i możliwości interpretacji głębokich sieci neuronowych w zastosowaniach związanych z autonomicznymi pojazdami. Jednym z głównych wkładów jest metoda sterowanego uczenia sieci, która usprawnia trening modelu poprzez wykorzystanie informacji zwrotnej z algorytmu wizualizującego uwagę sieci. Ponadto opracowano nowe architektury sieci neuronowych do szacowania cech geometrycznych z obrazów monokularnych. Architektury te uwzględniają w funkcjach kosztu modele 3D obiektów i ograniczenia geometryczne środowiska w celu zwiększenia dokładności lokalizacji punktów kluczowych, zapewniając predykcje zgodne z fizyczną strukturą obserwowanych obiektów i stabilne rozwiązanie nawet w scenariuszach z ograniczoną ilością danych treningowych.

Ponadto rozprawa bada metody szacowania niepewności predykcji poprzez sieci neuronowe. Dedykowana sieć neuronowa została zaprojektowana do estymacji niepewności współrzędnych punktów kluczowych 2D i 3D, oferując probabilistyczną miarę niepewności każdej predykcji. Dodatkowo, przedstawiony został potok przetwarzania do szacowania pozycji otaczających pojazdów przy użyciu obrazów monokularnych. Potok ten integruje algorytm Unscented Transform w celu propagacji niepewności estymowanych punktów kluczowych 2D i 3D, w celu uzyskania niepewności estymowanej pozycji pojazdu.

Proponowane metody zostały zweryfikowane na dwóch scenariuszach. Pierwszy scenariusz obejmował manewr dokowania do stacji ładowania przy użyciu elektrycznego autobusu miejskiego. W tej aplikacji system z powodzeniem szacował pozycję autobusu względem stacji ładowania, umożliwiając precyzyjne dokowanie z marginesem błędu mniejszym niż 30 cm z odległości ponad 30 metrów. Drugi scenariusz koncentruje się na estymacji pozycji otaczających pojazdów w środowisku miejskim. Dokładność szacowanej pozycji pojazdów została zweryfikowana wykorzystując zbiór danych ApolloCar3D, który zawiera obrazy pochodzące ze środowiska miejskiego. Zaproponowana metoda osiągnęła wyniki state of the art na wyżej wymienionym zbiorze danych.

Metody i ich eksperymentalna ewaluacja przedstawione w niniejszej rozprawie potwierdzają tezy badawcze: Po pierwsze, architektury, które ekstrahują i wizualizują interpretowalne cechy, mogą usprawnić trening sieci poprzez rozszerzenie istniejących zbiorów danych. Po drugie, sieci neuronowe są w stanie opisać niepewność wytworzonych cech geometrycznych. Po trzecie, wykorzystanie dostępnych modeli 3D obserwowanych obiektów ułatwia trening sieci do estymacji cech geometrycznych z obrazów 2D bez dokładnego etykietowania i znacznie poprawia dokładność wykrywania tych cech. Wreszcie, znajomość ograniczeń geometrycznych ze znanych modeli obiektów pomaga zmniejszyć liczbę fałszywych detekcji cech i zwiększa precyzję ich lokalizacji.

Acknowledgements

I would like to express my deepest appreciation to my supervisor, Piotr Skrzypczyński, for his unwavering support throughout my research. His knowledge, patience, understanding, and exceptional motivational skills have significantly contributed to my development as a researcher. Thank you for guiding me through this challenging journey.

I am also immensely grateful to my auxiliary supervisor, Michał Nowicki, for sharing an extensive amount of knowledge and for the invaluable discussions and advice that have enriched my research experience. Your expertise has been important in shaping my academic perspective and enhancing my work.

I extend my heartfelt thanks to my colleague, Krzysztof Ćwian, for the enriching discussions, steadfast support, and camaraderie throughout our doctoral studies. Your insights and companionship have been invaluable and made this journey both productive and enjoyable. I am truly grateful for having shared this path with you.

I extend my gratitude to my parents and sister for instilling in me a profound curiosity about the world and a relentless desire to pursue knowledge. Your unwavering support and encouragement have significantly shaped my personal growth. Thank you for inspiring me to explore and understand the complexities of the world around us.

I am deeply grateful to my wife and daughter for their constant support and the happiness they bring to my life. Thank you both for your understanding and patience when my studies demanded much of my attention. Your support and encouragement have been essential throughout this journey. Your presence has truly made every challenge more manageable and every success more meaningful.

Abbreviations

6DoF	**6** **D**egrees **o**f **F**reedom
A3DP-Abs	**A**verage **3D** **P**recision - **Abs**olute
ADAS	**A**dvanced **D**river **A**ssistance **S**ystem
AI	**A**rtificial **I**ntelligence
BFGS	**B**royden-**F**letcher-**G**oldfarb-**S**hanno
BRIEF	**B**inary **R**obust **I**ndependent **E**lementary **F**eatures
CAD	**C**omputer-**A**ided **D**esign
CDF	**C**umulative **D**istribution **F**unction
CNN	**C**onvolutional **N**eural **N**etwork
DBSCAN	**D**ensity-**B**ased **S**patial **C**lustering of **A**pplications with **N**oise
DGPS	**D**ifferential **G**lobal **P**ositioning **S**ystem
DNN	**D**eep **N**eural **N**etwork
ELU	**E**xponential **L**inear **U**nit
EV	**E**lectric **V**ehicle
EPnP	**E**fficient **P**erspective-**n**-**P**oint
FAST	**F**eatures from **A**ccelerated **S**egment **T**est
FNMPC	**F**ast **N**onlinear **M**odel **P**redictive **C**ontrol
FoV	**F**ield **o**f **V**iew
FPS	**F**rames **P**er **S**econd
FSOD	**F**ew-**S**hot **O**bject **D**etection
GAKN	**G**eometry-**A**ware **K**eypoint **N**etwork
GAN	**G**enerative **A**dversarial **N**etwork
GAT	**G**raph **A**ttention Ne**T**work
GCN	**G**raph **C**onvolutional **N**etwork
GCNConv	**G**raph **C**onvolutional **N**etwork **Conv**olution
GPS	**G**lobal **P**ositioning **S**ystem
IMU	**I**nertial **M**easurement **U**nit
IoU	**I**ntersection **o**ver **U**nion
KLD	**K**ullback-**L**eibler **D**ivergence
KNN	**K** **N**earest **N**eighbors
KSH	**K**eypoint **S**core **H**ead
LiDAR	**L**ight **D**etection **a**nd **R**anging
LSTM	**L**ong **S**hort-**T**erm **M**emory

MAML	**M**odel-**A**gnostic **M**eta-**L**earning
ML	**M**achine **L**earning
MLP	**M**ulti**l**ayer **P**erceptron
MPJPE	**M**ean **P**er **J**oint **P**osition **E**rror
MPV	**M**ulti-**P**urpose **V**ehicle
MPVPE	**M**ean **P**er **V**ertex **P**osition **E**rror
MRHKN	**M**ax **R**esolution **H**eatmap **K**eypoint **N**etwork
MSE	**M**ean **S**quared **E**rror
ORB	**O**riented FAST and **R**otated **B**RIEF
P3P	**P**erspective-**3**-**P**oint
PCA	**P**rincipal **C**omponent **A**nalysis
PCK	**P**ercentage of **C**orrect **K**eypoints
PHEV	**P**lug-in **H**ybrid **E**lectric **V**ehicle
PnP	**P**erspective-**n**-**P**oint
PROSAC	**Pro**gressive **Sa**mple **C**onsensus
PUT	**P**oznan **U**niversity of **T**echnology
RADAR	**Ra**dio **D**etection **a**nd **R**anging
RANSAC	**Ran**dom **Sa**mple **C**onsensus
ReLU	**R**ectified **L**inear **U**nit
RKN	**R**egression **K**eypoint **N**etwork
RMSE	**R**oot **M**ean **S**quared **E**rror
RNN	**R**ecurrent **N**eural **N**etwork
ROI	**R**egion **O**f **I**nterest
RPN	**R**egion **P**roposal **N**etwork
RTK	**R**eal-**T**ime **K**inematic
SBC	**S**olaris **B**us and **C**oach
SIFT	**S**cale-**I**nvariant **F**eature **T**ransform
SLAM	**S**imultaneous **L**ocalization **A**nd **M**apping
SUV	**S**port **U**tility **V**ehicle
SWIN	**S**hifted **Win**dow
UDP	**U**nbiased **D**ata **P**rocessing
UEH	**U**ncertainty **E**stimation **H**ead
UT	**U**nscented **T**ransform
VAE	**V**ariational **A**uto**e**ncoder
ViT	**V**ision **T**ransformer

Notation

$\mathbf{M}$	Faster R-CNN feature maps
$\mathbf{A}$	Faster R-CNN anchors representing different bounding box scales and aspect ratios
$\mathbf{R}$	Faster R-CNN weights for creating an attention heatmap
$\mathbf{H}$	Faster R-CNN attention heatmap
$\mathbf{P}(j,i)$	Faster R-CNN weight of the j-th feature contributing to the i-th anchor
$\mathcal{L}_{FRCNN}$	Faster R-CNN loss function
N_{cls}	Normalising factor for the classification loss
N_{reg}	Normalising factor for the regression loss
$\mathcal{L}_{cls}(p_i, p_i^*)$	Faster R-CNN classification loss function
$\mathcal{L}_{reg}(t_i, t_i^*)$	Faster R-CNN regression loss function
p_i	Predicted probability of the i-th anchor being an object
p_i^*	Ground-truth label of the i-th anchor being an object
t_i	Predicted bounding box
t_i^*	Ground-truth bounding box
$\mathcal{L}_\delta$	Huber Loss function
λ_{FRCNN}	Weight balancing the regression loss relative to the classification loss
$\mathcal{L}_{MSEkpts}$	Mean Squared Error (MSE) loss for keypoints
$\mathbf{p}^{2d}$	Ground truth 2D coordinates of the i-th keypoint on the image
$\hat{\mathbf{p}}_i^{2d}$	Estimated 2D coordinates of the i-th keypoint on the image
$\|\cdot\|_2$	Euclidean norm (L2 norm)
$\mathcal{L}_{RKN}$	Loss function for the Regression Keypoint Network (RKN)
K_i	Pixels cluster in the raw heatmap
S_i	Confidence score for cluster K_i
$I(\mathbf{x})$	Intensity value at pixel location $\mathbf{x}$
$\mathbf{x}$	Pixel location represented as $\mathbf{x} = [u, v]$
$\mathbf{c}_i$	Final keypoint location, computed as cluster's center of mass
$\mathcal{L}_{BCE}$	Binary Cross-Entropy Loss
π	Camera projection function
$\mathbf{K}$	Camera intrinsics matrix
$\mathbf{T}$	Rigid transformation matrix
$\mathbf{p}^{3d}$	Ground truth 3D coordinates of the i-th keypoint
$\hat{\mathbf{p}}_i^{3d}$	Estimated 3D coordinates of the i-th keypoint
$\tilde{\mathbf{p}}_i^{2d}$	Projected i-th 3-D keypoint on image

$\mathbf{T}^*$	Optimal rigid transformation matrix minimising the reprojection loss
$\mathcal{L}_{repr}$	Reprojection loss value for the optimal transformation $\mathbf{T}^*$
$\mathcal{L}_{MSE}$	Mean Squared Error loss for heatmaps
m_{ij}	j-th pixel of the ground truth heatmap corresponding to the i-th point
$\hat{m_{ij}}$	j-th pixel of the predicted heatmap corresponding to the i-th point
$\mathcal{L}_{\text{reprojection}}$	Reprojection loss function
$\lambda_{\text{reprojection}}$	Scaling weight for the reprojection loss
$\|\cdot\|_1$	Manhattan norm (L1 norm)
$\mathbf{L}_{2D}$	Lower triangular matrix with positive diagonal entries
$\mathbf{\Sigma}$	Covariance matrix representing uncertainty
$\mathcal{L}_{unc}$	Gaussian Log-Likelihood Loss function
$\mathcal{L}_{\text{MPVPE}}$	Mean Per Vertex Position Error loss
λ_{MPVPE}	Scaling weight for the Mean Per Vertex Position Error loss
$\mathcal{L}_{\text{Dim}}$	Dimension discrepancy loss
λ_{Dim}	Scaling weight for the dimension discrepancy loss
$\mathcal{L}_{\text{KLD}}$	Kullback-Leibler Divergence loss
λ_{KLD}	Scaling weight for the Kullback-Leibler Divergence loss
$\hat{\mathbf{v}}_i$	Estimated position of the i-th vertex
$\mathbf{v}_i$	Ground truth position of the i-th vertex
$\hat{l}$, $\hat{w}$, $\hat{h}$	Estimated length, width, and height of the car
l, w, h	Ground truth length, width, and height of the car
μ	Mean of the latent variable
σ	Standard deviation of the latent variable
$\mathcal{L}_{\text{Kpts3D}}$	Keypoints 3D loss function
λ_{Kpts3D}	Scaling weight for the Keypoints 3D loss
(f_u, f_v)	Focal lengths of the camera along the u and v axes
(c_u, c_v)	Central point of the camera's image
θ	The yaw angle of the charger roof with respect to the camera coordinate system
γ	The pitch angle of the charger roof with respect to the camera coordinate system
r	Actual height of the charger
s	Actual width of the charger
$\tilde{r}$	Height of the charger's image projection in pixels
$\tilde{s}$	Width of the charger's image projection in pixels
I_{v1}	Lower edge of the charger's projection relative to the image's central axis
I_{v2}	Upper edge of the charger's projection relative to the image's central axis
I_u	Position of the charger's center on the image along the u axis
α	Angle calculated using the upper edge of the charger's projection
β	Angle difference between the upper and lower edges of the charger's projection
γ_{ref}	A parameter of Reprojection-based Pose Renement defining error acceptance threshold
n	Total number of predicted keypoints
$n_{correct}$	Number of correctly estimated keypoints
(x_i, y_i, z_i)	Ground truth coordinates of the i-th keypoint in 3D space
$(\hat{x}_i, \hat{y}_i, \hat{z}_i)$	Predicted coordinates of the i-th keypoint in 3D space
$\mathcal{L}_{stage1}$	Loss function for stage 1 of the model training

w_{repr}	Weight for the reprojection loss
$\mathcal{L}_{reprojection}$	Reprojection loss
$\mathcal{L}_{heatmap3Dxy}$	Mean Squared Error loss for 3D heatmap in the xy plane
$\mathcal{L}_{heatmap3Dxz}$	Mean Squared Error loss for 3D heatmap in the xz plane
$\mathcal{L}_{heatmap2D}$	Mean Squared Error loss for 2D heatmap
$\mathcal{L}_{stage2}$	Loss function for stage 2 of the model training
$\mathcal{L}_{uncertainty2D}$	Loss function for 2D uncertainty estimation
$\mathcal{L}_{uncertainty3D}$	Loss function for 3D uncertainty estimation
$\mathcal{L}_{keypoint\_score}$	Mean Squared Error loss for keypoint scoring
χ	Unscented Transform sigma point
λ_χ	Scaling factor for sigma points
α_χ	Parameter controlling the spread of the sigma points around the mean
β_χ	Parameter of Unscented Transform algorithm for capturing higher-order moments
k_χ	Secondary scaling parameter influencing the distance of sigma points from the mean
$\mathbf{m_x}$	Mean of the estimated points
m_y	Mean of the transformed sigma points
$\mathbf{C_x}$	Covariance matrix of the estimated points
$\mathbf{C_y}$	Covariance matrix of the transformed sigma points
Y_i	Sigma points after passing through the nonlinear transformation
w_i^c	Weights associated with i-th sigma point for covariance calculation
${}^E\mathbf{T}_F$	Pose of the bus's front axis with respect to the charger
${}^C\mathbf{T}_E$	Pose of the electric charger in the camera coordinate system
${}^C\mathbf{T}_F$	Pose of the camera coordinate system with respect to the front axis of the bus
c_{trans}	Translation error metric
c_{rot}	Rotation error metric
$\mathbf{t}_{gt}$	Ground truth translation vector
$\hat{\mathbf{t}}$	Estimated translation vector
$\mathbf{q}_{gt}$	Ground truth rotation quaternion
$\hat{\mathbf{q}}$	Estimated rotation quaternion
δ_t	Acceptance threshold for translation error in A3DP metric
δ_{rot}	Acceptance threshold for rotation error in A3DP metric

Contents

Chapter 1

Introduction

1.1 Motivation

Autonomous vehicles represent a revolutionary development in transportation technology that has the potential to transform the way we travel and live. These vehicles can navigate without human intervention and offer numerous benefits such as increased safety, reduced traffic congestion, and improved accessibility [69].

Several companies are at the forefront of autonomous vehicle development, each bringing unique innovations. Tesla, Inc. is a pioneering automotive company known for its electric vehicles. Founded in 2003 by Elon Musk among others, Tesla has revolutionised the industry with products such as Model S, and Model 3, pushing the boundaries of sustainable transportation. Google's Waymo, originally a project within Google, has become an independent company. In 2021, they launched a robot taxi service in San Francisco. General Motors also plays a significant role in this field through Cruise Automation, which was founded in 2013 and later acquired by General Motors in 2016. Cruise focuses on developing autonomous driving technology for the Chevrolet Bolt, employing Light Detection and Ranging (LiDAR) sensors from Velodyne. In January 2020, Cruise unveiled the Origin, a fully autonomous car with no steering wheel or pedals, designed for its ride-sharing service and capable of Level 5 autonomous driving. The Origin will have a modular design with two rows of seats facing each other, a lifespan of one million miles, and a hybrid sensor assembly called Owl that combines both cameras and radar [92]. TIER IV, an innovator in open-source autonomous driving technology, has launched the L4 RIDE initiative in Japan, aimed at addressing driver shortages and promoting regional development through autonomous bus services. After securing Level 4 certification in Greater Tokyo in October 2023 and conducting proof of concept tests in several locations, TIER IV is now focused on achieving nationwide commercialization of these technologies [98]. Concurrently, Mercedes-Benz has marked a breakthrough by becoming the first international automaker to receive approval for Level 4 automated driving tests on urban roads and highways in Beijing, aiming to integrate these vehicles into regular traffic for a range of complex maneuvers without human intervention.

These developments represent a substantial push towards the integration of highly automated vehicles into everyday traffic scenarios, enhancing the safety and efficiency of urban transportation systems [64].

A core system in all autonomous cars is the perception system [110]. This is a network of sensors and software designed to emulate human sensory and cognitive functions for driving. This system typically integrates various types of sensors, each serving a distinct purpose. The primary sensors are cameras, which capture detailed visual information, such as traffic signals, road signs, cars, and pedestrians. LiDAR sensors provide 3D mapping capabilities, generating high-resolution point clouds of the cars surroundings. These are essential for understanding complex environments and detecting obstacles. Radio Detection and Ranging (RADAR) complements these by offering robust performance in adverse weather conditions and measuring the speed and distance of objects with high precision. Additionally, ultrasonic sensors are employed for close-range detection tasks, like parking assistance. The data generated by these sensors is fed into the cars central processing unit, where algorithms analyse and interpret the information. This allows the vehicle to make informed decisions about navigation, obstacle avoidance, and speed control. The integration of diverse sensors composing a perception system allows the construction of safe and reliable autonomous vehicles.

Modules of autonomous vehicle perception systems based solely on cameras offer several advantages, primarily due to their simplicity and cost-effectiveness compared to setups that require multiple types of sensors. Moreover, being passive sensors, cameras do not interfere with other systems [120]. Cameras can capture detailed visual information similar to the human eye, providing necessary data for a wide range of tasks such as object detection, traffic sign recognition, and pose estimation. These visual systems can leverage advancements in computer vision and deep learning to interpret complex scenes and make informed driving decisions. Furthermore, the reliance on cameras reduces hardware costs and simplifies the integration and maintenance of the system, making it more accessible and potentially accelerating broader adoption. Despite some challenges in low-light and adverse weather conditions, ongoing improvements in camera technology and image processing algorithms continue to enhance their reliability, underscoring their potential as a primary or supplementary technology in autonomous driving systems.

One of the possible applications of camera-based systems in autonomous driving technology is the estimation of the pose of surrounding objects and vehicles. These systems determine the position, orientation, and velocity of other vehicles relative to the autonomous car. Accurate pose estimation allows the autonomous vehicle to anticipate the future movements of nearby cars, which is required for making informed decisions about maneuvers such as changing lanes or maintaining safe distances. These camera-based systems employ computer vision algorithms, often based on deep learning, to interpret single or multiple consecutive frames and predict the vehicle's position and orientation relative to the camera. As camera resolution and processing capabilities continue to improve, the effectiveness of camera-only systems in pose estimation also enhances, making them a viable solution for understanding dynamic driving environments [31].

The estimation of 3D information from 2D images represents a well-known problem in computer vision, particularly challenging due to its inherently ill-constrained nature [33]. When translating data from a single two-dimensional image into three-dimensional estimations, the lack of depth

information makes the task underdetermined without additional data or constraints. This implies that multiple possible 3D configurations can result in the same 2D image, leading to ambiguities in depth perception and object recognition [33].

While there has been significant progress in the development of autonomous vehicles, numerous challenges remain to be addressed. One of the most significant challenges is ensuring the safety and reliability of these vehicles in different environments and situations. This requires the development of algorithms that can explain the decisions made by neural networks and measure the confidence of their predictions. In the context of autonomous driving, explainable machine learning (ML) offers insights into the decision-making processes of Articial Intelligence (AI) models. The principal benefit of integrating explainability into ML systems is the capacity to interpret and justify the actions and decisions made by autonomous vehicles. This can be used to ensure that the models not only perform optimally but also adhere to safety regulations and ethical standards. Moreover, explainable machine learning enables the assessment and management of uncertainty, which is an important element in maintaining robust autonomous systems where unexpected environmental variables frequently occur. Furthermore, explainable machine learning approaches can offer adaptability through self-evolving mechanisms that learn from new, unforeseen situations, thereby enhancing the safety and reliability of autonomous systems. This adaptability is critical, especially in scenarios where conventional deep learning methods may misinterpret unfamiliar scenes with high confidence, potentially leading to disastrous consequences. Thus, explainable models in autonomous driving not only need to provide high performance metrics like accuracy and F1 score but also to provide reliable explainability and safety mechanisms [90] [4].

Another challenge is the strong dependence of deep learning models on large training datasets. The algorithms proposed in the field of computer vision are often overfitted to achieve the best results on a given test set. The computer vision benchmarks allow a fair comparison of different algorithms but typically represent only a small proportion of the cases occurring in the real world. When attempting to apply them in robotics, in an application-specific environment, there is a significant drop in the quality of the model when the input data originates from target use cases. A common practice to cope with the aforementioned problems is fine-tuning a base model pretrained on benchmark datasets using data from the target environment. Conducting experiments and gathering training data using a robot or a vehicle is often costly and logistically challenging. Furthermore, access to the target work environment (e.g. public roads, production plants) is usually constrained. This severely limits the possibility of collecting the requisite amount of training data for a given task.

In conclusion, explainable machine learning algorithms are essential in ensuring the high accuracy of autonomous driving systems. Transparency is crucial for enhancing safety and fostering trustworthiness. Furthermore in order to scale up the implementation of these methods for autonomous cars, it is also necessary to reduce the cost of collecting and labelling training data. This PhD thesis aims to contribute to this research by proposing innovative solutions that can balance the trade-off between accuracy and interpretability to improve the transparency and trustworthiness of autonomous driving systems.

1.2 Problem statement

The training of neural networks with limited datasets presents a significant challenge, primarily due to the risk of overfitting. When a neural network is trained on a small amount of data, it may learn to perform exceptionally well on its training examples but fail to generalise to new, unseen data. This lack of generalisation can severely hinder the model's practicality and reliability when deployed in real-world scenarios. To cope with such problems, researchers often resort to techniques such as data augmentation, transfer learning, or synthetic data generation to artificially enlarge the training dataset and help improve the model's robustness and accuracy. Nevertheless, these methods can only partially mitigate the limitations posed by small datasets. Consequently, there is a continued need to identify innovative ways to train effective models with limited data, which remains an area of intensive research in the field of machine learning [5].

One potential solution to this problem may be the extraction and visualisation of meaningful intermediate features generated by neural networks. The aforementioned features provide insight into the internal workings of the neural network, demonstrating how the model processes and transforms input data through various stages of abstraction. This capability not only enhances the understanding of the learning and decision-making processes of these models but also opens up new possibilities for improving the learning process itself. For example, the visualisation of these features enables researchers to identify which aspects of the data are being emphasised or ignored, thereby facilitating targeted adjustments to the model architecture or training process. Furthermore, the ability to extract intermediate features has significant implications for data augmentation. The generation of new data based on visualisations of internal network states allows for the expansion of the training dataset, thereby providing the model with more accurate examples for learning. This is particularly advantageous in contexts where data is scarce or costly to obtain.

Measuring the score or confidence of neural network predictions is another important challenge, as it directly impacts decision-making processes and system reliability. Typically, neural networks output probabilities or raw scores that may not directly correlate with the actual likelihood of correctness. This issue is of particular significance in high-risk domains such as healthcare and autonomous driving, where the cost of incorrect predictions can be extremely high [90].

Furthermore, neural networks, particularly deep learning models, may exhibit an excessive degree of confidence in their predictions, even when they are erroneous. This phenomenon of overconfidence can be attributed to the training procedure, which typically defines labels as discrete categories and does not represent confidence in the training data. Consequently, the development of methodologies for the accurate estimation of the confidence of neural network predictions and the assurance that these estimates align with the true values is important for ensuring the trustworthiness and safety of AI systems [75].

The second considered task is an estimation of the pose of surrounding objects and vehicles. This task is approached using a monocular camera and is divided into two distinct scenarios. The first involves estimating the pose of known object with predefined dimensions and shape and the viewpoints are limited. The second scenario deals with the estimation of vehicles whose

shapes can vary widely and are not known beforehand. This requires the pose estimation system to adapt to a broader range of object geometries and to operate effectively even when detailed prior information is not available.

The pose estimation using a monocular camera presents a number of unique challenges, the most significant of which is the lack of depth information that is more readily available in systems that utilise multiple cameras or depth sensors. A monocular setup, which relies on a single camera, must infer the three-dimensional position and orientation of objects from two-dimensional images. This depth ambiguity can give rise to significant complexities in accurately determining the distance of objects from the camera, as well as their spatial orientation.

The estimation of pose in monocular vision is contingent upon the interpretation of visual cues within the image, including shadows, texture, perspective, and scaling, in order to infer depth. While these cues can provide valuable information, they often require the use of complex and computationally intensive algorithms in order to be interpreted effectively. In some scenarios, it is possible to use a known 3D model of the object under consideration, but often these data are not known and must be inferred from the input images. Furthermore, the precision of monocular pose estimation may be diminished in conditions of poor lighting or environments with minimal texture or other helpful visual features. Consequently, while the simplicity and lower cost of monocular camera systems are advantageous, these benefits come with significant challenges that must be addressed to ensure reliable and accurate pose estimation [120].

Algorithms that solves the Perspective-n-Point (PnP) problem, which are employed to estimate the pose of an object based on a set of 3D-to-2D point correspondences, are confronted with a number of challenges that can impact their performance and accuracy. One significant challenge is the sensitivity to noisy data. In practical situations, the 2D points identified in images may be affected by several factors, including inadequate lighting, occlusions, and imperfections in the feature detection techniques employed. The inaccuracies in the 2D points can result in significant errors in the computed pose, thereby rendering the solution less reliable.

Furthermore, the efficacy of PnP algorithms is inherently dependent on the quality of the initial guess, particularly in the context of non-linear optimisation-based methodologies. In the absence of a sufficiently accurate initial estimate, these algorithms tend to converge on local minimum, resulting in inaccurate pose estimations. This requirement constrains their efficacy in fully automated systems where the provision of initial estimates by humans is impractical.

Furthermore, the robustness of these algorithms is limited in cases where the number of point correspondences is minimal or when the points are collinear or coplanar. This makes the pose estimation problem underdetermined. In such circumstances, the algorithms encounter difficulties in identifying a unique and accurate solution, which can result in unstable or ambiguous pose estimations. This underscores the necessity for the development of robust methodologies that can effectively cope with a wide range of scenarios and conditions.

Pose estimation leverages geometric correspondences and the properties of camera projection geometry, setting it apart from other tasks like image classification or object detection. While data-driven methods have been successful in the latter areas, incorporating model-driven strategies into pose estimation can lead to enhanced performance. This integration of data-driven and

model-driven approaches offers a promising hybrid strategy, which can combine the reliability of empirical data with the precision of geometric modeling, thereby improving the accuracy and robustness of pose estimation systems.

The technique of learning from geometric priors is a specialised method within the domains of machine learning and computer vision. It involves the incorporation of known geometric constraints or characteristics, such as the shape, size, and spatial relationships of objects, into the learning process. This approach is useful in tasks where the geometry of the objects or environment plays an important role, especially in the contexts of 3D reconstruction and pose estimation.

In human pose estimation, the knowledge of the human body's structure enables the prediction of more realistic body poses and the avoidance of physically impossible configurations. Similarly, in the context of pose estimation for autonomous driving, knowledge of the geometry of cars and other road infrastructure objects can enhance the accuracy of the predictions made.

In the context of training datasets preparation, geometric priors can be applied for the reconstruction of three-dimensional features, such as shape, dimensions, and the position of a given point in three-dimensional space. The extraction of such information from 2D images frequently necessitates the deployment of extensive and precise manual labelling, a process that is both time-consuming and susceptible to errors and inconsistencies. Nevertheless, the availability of accurate three-dimensional models of observed objects offers new possibilities for the preparation of datasets and the training of machine learning algorithms. The 3D models contain detailed geometric information that can be projected onto 2D planes, simulating various perspectives. By leveraging these projections, a rich set of training data can be generated that inherently contains geometric constraints, thereby reducing the dependency on exact labelling.

Despite the advantages, several challenges must be addressed when applying geometric priors and 3D models in training dataset preparation. One significant challenge is the computational complexity involved in processing and projecting 3D models onto 2D planes. This task necessitates the utilisation of substantial computational resources and the deployment of efficient algorithms to oversee and manipulate voluminous datasets, which can be resource-intensive and time-consuming. Furthermore, ensuring the alignment and calibration of 3D models with real-world 2D images represents a significant challenge. In order for the projections to be effective for training purposes, it is necessary for them to accurately represent the observed objects under a variety of conditions. This requires the use of precise calibration techniques and robust alignment algorithms.

The incorporation of geometric priors into machine learning models enables the leveraging of human understanding of physical and spatial relationships, thereby enhancing the efficiency of the models, reducing the necessity for extensive datasets, and frequently improving performance in tasks where geometry is a key factor. This integration of geometric knowledge with learning algorithms represents a potent blend of model-based and data-driven approaches, frequently resulting in enhanced predictive performance and reduced cost of training datasets preparation [87], [105].

Based on the above considerations, it is possible to formulate a research thesis consisting of four parts:

- Deep learning architectures that allow us to extract and visualise meaningful intermediate features make it possible to guide learning by augmenting the existing data sets.
- Deep learning architectures allow us to describe the uncertainty of the geometric features produced by the network.
- Using the available 3D models of observed objects makes it possible to learn geometric features from 2D images of these objects without exact labelling, and improves geometric feature detection from 2D images of these objects.
- The knowledge of the geometric constraints stemming from known object models allows a deep learning architecture to decrease the number of falsely detected features and increases the accuracy of feature location.

1.3 Content of the thesis

The following chapters (2-5) discuss tasks related to visual perception. When integrated into a single processing pipeline, these submodules will allow for the creation of a module to describe the environment of an autonomous car using just a single camera with uncertainty assessment trained using limited datasets.

Chapter 2 introduces a method for visualising the object detection network's spatial areas of interest. This method facilitates a more comprehensive understanding and optimisation of the model, by highlighting which regions within an image the network prioritises. Furthermore, the chapter outlines a guided learning procedure that enhances the network's performance in scenarios where training data is limited, by refining the training process based on specific feedback from the aforementioned visualisation method.

Chapter 3 addresses the challenge of estimating key points in an image. Three novel network architectures, designed with the specific objective of estimating points at electric bus charging stations, are presented. A novel loss function is introduced, which facilitates the training of the network. Furthermore, the chapter presents a method for estimating the uncertainty of network predictions.

Chapter 4 presents a neural network architecture for estimating 3D coordinates of characteristic points from a single image, trained using a reprojection loss function. A novel methodology for the automated acquisition of ground truth 3D keypoint locations from a dataset comprising images, 2D keypoint labels, and vehicle Computer-aided Design (CAD) models is presented. Finally, two neural network architectures for encoding and estimating the shape of cars are presented.

Chapter 5 outlines a pipeline for car pose estimation, starting with the 2D keypoint detection on images using a neural network presented in Chapter 3 and estimating corresponding 3D coordinates using an algorithm presented in Chapter 4. The pose estimation is conducted through a custom PnP algorithm, which leverages the 2D-3D point correspondences. Furthermore, the

pipeline incorporates an uncertainty estimation mechanism, which propagates the uncertainties from the detected 2D keypoints and the estimated 3D points to provide a measure of the uncertainty associated with the pose estimates.

Chapter 6 demonstrates the practical applications of the techniques outlined in Chapters 2 to 5, illustrating their effectiveness through case studies and real-world deployments. An ablation study is also included, which is a methodical assessment of the effects of different components of the proposed methods. This emphasises both the singular and combined effects of these components on the overall system performance. Furthermore, this chapter presents a comprehensive comparison of the findings with the existing state-of-the-art (SOTA) technologies, demonstrating the advancements and enhancements introduced by the proposed methods to the domain of perception systems for autonomous vehicles.

Chapter 7 provides a summary of the dissertation and includes a section on conclusions, which outlines the contributions made to the field of machine learning in autonomous driving and highlights fragments where the particular research theses are proven. Furthermore, this chapter outlines prospective directions for future research.

1.4 Projects and publications

The research presented in this thesis, conducted from 09.2018 to 12.2020, was made possible by financial support from the project: "Advanced Driver Assistance System (ADAS) for precision maneuvers with single-body and articulated urban buses" funded by the National Centre for Research and Development, where the author worked as a Research Assistant.

Some of the results presented in this dissertation have been previously published in the following journal articles:

- T.Nowak, M. R. Nowicki, P. Skrzypczyński, Vision-based positioning of electric buses for assisted docking to charging stations, International Journal of Applied Mathematics and Computer Science, vol. 32, no. 4, p. 583-599, 2022, IF_{2022}:1.79

Selected results have been also presented during the following conferences:

- T.Nowak, M. R. Nowicki, K. Ćwian, P. Skrzypczyński, How to Improve Object Detection in a Driver Assistance System Applying Explainable Deep Learning, 2019 IEEE Intelligent Vehicles Symposium, 30th IEEE Intelligent Vehicles Symposium, 9-12.06.2019, Paris, France, p. 226-231
- T.Nowak, M. R. Nowicki, K. Ćwian, P. Skrzypczyński, Leveraging Object Recognition in Reliable Vehicle Localization from Monocular Images, Automation 2020: Towards Industry of the Future: Proceedings of Automation 2020, Automation 2020, 18-20.03.2020, Warsaw, Poland, p. 195-204

- T.Nowak, P. Skrzypczyński, Geometry-Aware Keypoint Network: Accurate Prediction of Point Features in Challenging Scenario, Proceedings of the 17th Conference on Computer Science and Intelligence Systems, 17th Conference on Computer Science and Intelligence Systems FedCSIS 2022, 4-7.09.2022, Sofia, Bulgaria, pp. 191-200
- T.Nowak, Accurate Camera Pose Estimation from Learned Point Features: A Case Study, Proceedings of the 3rd Polish Conference on Artificial Intelligence PP-RAI'2022, 3rd Polish Conference on Artificial Intelligence PP-RAI'2022, 25-27.04.2022, Gdynia, Poland, p. 98-102
- T.Nowak, P. Skrzypczyński, A New Approach to Learning of 3D Characteristic Points for Vehicle Pose Estimation, Progress in Polish Artificial Intelligence Research 4, 4th Polish Conference on Artificial Intelligence PP-RAI'2023, 24-26.04.2023, Łódź, Poland, p. 389-394
- T.Nowak, P. Skrzypczyński, A Neural Network Architecture for Accurate 4D Vehicle Pose Estimation from Monocular Images with Uncertainty Assessment, Neural Information Processing: 30th International Conference, ICONIP 2023, Changsha, China, November 2023, 2023, Proceedings, Part VIII, 30th International Conference on Neural Information Processing, ICONIP 2023, 20-23.11.2023, Changsha, China, p. 396-412
- T.Nowak, P. Skrzypczyński, Precision Vehicle Pose Estimation with Uncertainty-aware Neural Network, Walking Robots into Real World, 27th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines, CLAWAR 2024, 4-6.9.2024, Kaiserslautern, Germany, p. 22-33

Chapter 2

Object detection

2.1 Introduction

Object detection is an important task in the context of autonomous driving, enabling vehicles to perceive and understand their surroundings in real-time [39]. This technology involves the identification and localisation of objects within the vehicle's environment, including pedestrians, other vehicles, traffic signs, and obstacles. Advanced deep learning models, such as convolutional neural networks (CNNs), are employed to process data from cameras in order to accurately and rapidly detect objects. The accurate detection of objects is of essential importance for the safe navigation of autonomous vehicles. Such vehicles must be able to avoid collisions, adhere to traffic regulations, and provide a smooth and secure driving experience.

Despite their efficacy in detecting objects in the real world, traditional object detection neural networks employed in autonomous vehicles are often undermined by a lack of transparency in the decision-making processes. In the context of autonomous driving, where safety and reliability are of high importance, the interpretability of these networks becomes a critical issue [90]. An essential aspect of understanding the functionality of an object detection network is the capacity to identify the rationale behind its decisions. This enables the diagnosis of errors, enhancement of system reliability, and the safeguarding of passengers and pedestrians. To address this issue, interpretable networks are developed, incorporating features that allow users to comprehend the reasoning behind each decision [124]. This includes techniques for visualising feature maps, attention mechanisms, or layer activations, which provide insight into the focus of the network when making decisions. Nevertheless, the development of interpretable object detection networks presents its own set of challenges, particularly in balancing computational complexity with interpretability.

Another challenge for object detection in autonomous driving is the dependence on large, annotated datasets [27], [58], [91]. In the field of machine learning, particularly in object detection, this traditional reliance has been a cornerstone of developing accurate models. However, in many real-world applications, including those in specialised fields or emerging areas of technology, access to such comprehensive datasets can often be limited. The limited availability of data can

result in models that are either overfitted or not sufficiently trained to handle the variability and unpredictability of real-world scenarios. Consequently, there is an increasing demand for methodologies capable of effectively using limited data to train models that are both accurate and generalisable. This constraint necessitates the development of alternative training strategies that can effectively work with smaller or less detailed datasets while still maintaining high performance. One potential solution to the aforementioned challenges is guided learning [28]. This approach is increasingly recognised as useful in the training of object detection networks, especially when the available training datasets are limited or do not comprehensively represent the environment in which the network is expected to function. Guided learning involves incorporating additional training examples, which may be synthetically generated or selectively curated, to aid the network in achieving better generalisation and reducing the risk of overfitting. In autonomous driving, the diversity and unpredictability of scenarios can be considerable. Guided learning can help train object detection models on a broader range of situations, including those that are less common or more complex, which might not be adequately represented in the initial dataset. The process of guided learning typically involves analysing the networks performance on a validation set and identifying specific weaknesses or biases in its detection capabilities. In the next step, improving the network's performance leverages incorporating carefully selected examples into the training set, which emphasise the challenging conditions. This addresses the identified weaknesses directly [20].

The problem addressed in this chapter involves the development of a neural network designed for the practical problem of detecting electric city bus charging stations. This network is a component of an ADAS aimed at aiding drivers to dock their vehicles at charging stations. A significant challenge is the lack of publicly available datasets that include images concerning charging stations, which are required for training robust detection models.

Moreover, the datasets currently accessible have been gathered in environments that differ from the target operational settings where the ADAS will be deployed. This discrepancy introduces additional complexities in adapting the neural network to perform effectively in real-world conditions where it is expected to function. Therefore, the development and validation of this network must consider these limitations, necessitating innovative approaches to overcome the absence of directly applicable training data. This situation underscores the need for a tailored solution that can learn from limited and non-representative data while maintaining high performance in diverse and specific deployment environments.

A distinctive feature of the proposed approach is the integration of advanced visualisation techniques. These techniques are not merely interpretive tools; they play an instrumental role in the model training process. By identifying and highlighting objects that are frequently misclassified or overlooked by the object detector, this method provides an invaluable feedback mechanism. This feedback enables the model to be improved in a targeted manner, with a particular focus on addressing its weaknesses and enhancing its accuracy and reliability in an iterative process. This chapter aims to effectively bridge the theoretical aspects of machine learning with their practical applications in the field of autonomous driving. It provides a detailed examination of the proposed methodology, covering the fundamental principles, the implementation steps, and the innovative use of visualization techniques. This investigation aims to contribute to the field

of autonomous vehicle technology, particularly in the development of robust and interpretable object detection systems that can be trained effectively with limited training data.

In this chapter, the following contributions to the field of guided learning applications for object detection are presented:

- A method for continuous visualization of Faster R-CNN's spatial areas of interests
- A guided learning procedure that improves performance when the training data is scarce

2.2 Related work

The evolution of object detection models using convolutional neural networks can be broadly categorised into one-stage and two-stage detectors. Each of these categories has its unique attributes and advancements. One-stage detectors, exemplified by models such as YOLO [83] [84] [85] [104], are known for their speed and efficiency. YOLO represents a pioneering model in this category, integrating the detection process into a single network, thereby enabling the model to predict object classes and locations in a single evaluation. In contrast, CenterNet [21] takes a different approach by representing objects as points rather than bounding boxes. Two-stage detectors, such as Fast R-CNN [29] and Faster R-CNN [86], represent another approach. These models divide the detection process into two stages. Initially, they generate region proposals, and subsequently, they classify and refine these proposals. This method, while generally slower than one-stage detectors, tends to offer higher accuracy and has been instrumental in pushing the boundaries of object detection performance. In more recent times, transformer-based detectors such as SWIN [59] and ViT [19] have emerged, integrating the transformer architecture, originally designed for natural language processing [100], into visual object detection.

Few-shot object detection (FSOD) [5] is an emerging area in computer vision that addresses the challenge of detecting objects from a limited number of examples. Traditional object detection models require large annotated datasets to enable effective training. However, this is not always a viable option due to the associated costs and efforts involved in data collection and annotation. FSOD aims to overcome this limitation by adapting models to recognise new objects from only a few training samples. A significant contribution to this field is the adaptation of meta-learning techniques, which prepare a model to learn quickly from limited data during the training phase. Techniques such as model-agnostic meta-learning (MAML) [118] have been particularly influential, as they allow the model to fine-tune effectively from minimal examples. Another approach involves the use of feature-reuse strategies, whereby a pre-trained model on a large dataset is adapted to new classes with minimal examples. This is achieved through the utilisation of transfer learning, which serves to reduce the necessity for extensive retraining [117]. These methodologies have led to the enhancement of deep learning models in terms of efficiency and adaptability to diverse real-world scenarios, where data scarcity is a common issue.

Object detection models also have applications in the field of autonomous driving. For example, the SqueezeDet model [112] features a fully convolutional network structure that integrates the computation of bounding boxes and class probabilities into a single forward pass, significantly

improving speed and model compactness while maintaining high accuracy on benchmarks such as KITTI [27].

The YOLOv4-5D model [12] demonstrates how changes to the network architecture, such as incorporating deformable convolution and an optimised network pruning algorithm, can improve detection accuracy while achieving real-time performance metrics on vehicular computing platforms. This model is particularly notable for improving the accuracy of detection of small objects through improvements in feature fusion.

A slightly different scenario is presented in [42], where vehicles are detected from traffic surveillance camera video. The challenges associated with different scales and occlusions have led to the development of multi-scale detection methods. These methods integrate additional prediction layers into conventional frameworks such as Yolo-v3, using techniques such as spatial pyramid pooling to improve the robustness and accuracy of vehicle detection under challenging conditions.

Neural networks have demonstrated remarkable performance in the domain of object detection. However, for their broader application in the autonomous driving field, it is essential to develop methods that enhance the interpretability of their decisions. Early efforts to visualise the activation of convolutional neural networks included the introduction of the deconvolution concept, as outlined in the work of Zeiler [121]. This was followed by an approach influenced by automated image captioning techniques, which were used for generating textual explanations within the realm of autonomous driving [9]. More recently, the study by Kim and Canny [40] investigated visual attention maps, akin to those utilised in this chapter, to pinpoint image areas that directly affected the steering decisions made by a CNN-controlled vehicle. Furthermore, research in a similar vein to ours [81] has demonstrated how to identify errors in object detection performed by neural networks through a comparative analysis of pairs of similar images. Kim Jung Uk [41] proposed a Spatial Relation Reasoning (SRR) framework that employs a Graph Convolutional Network to identify spatially related groups of meaningful image regions. In their paper, Wu et al. [114] present a method for integrating top-down grammar models with bottom-up convolutional networks for the purpose of learning two-stage object detection models that are qualitatively interpretable.

2.3 Proposed solution

2.3.1 Attention visualization

To enhance the performance of the Faster R-CNN in object detection, it is suggested to acquire insights into which areas of the image (and consequently, into which visible object types) receive the most focus during the detection phase. The hypothesis is that by identifying potential discrepancies with the targeted object, appropriate negative examples can be introduced into the training process to boost overall accuracy. Furthermore, the objective is to minimise the occurrence of false positives in detection, as any incorrect identification could be costly.

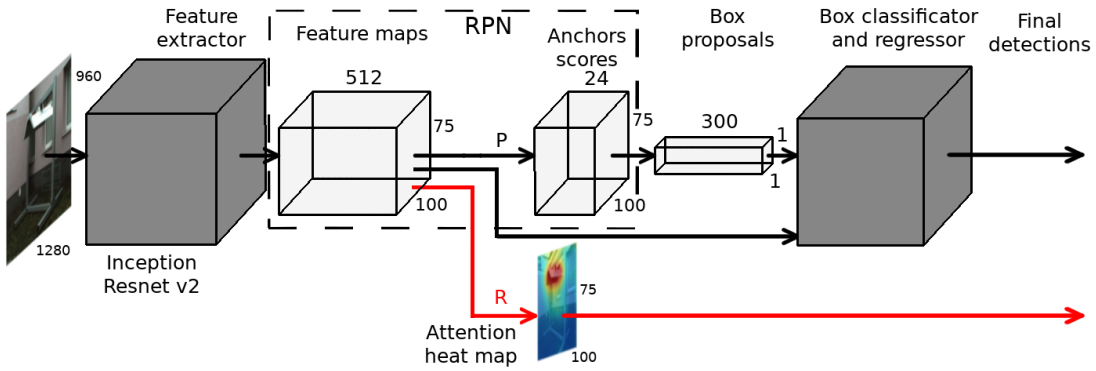

FIGURE 2.1: Processing pipeline of the Faster R-CNN consisting of feature extractor (Inception Resnet v2) generating feature maps that are used at all processing steps. The proposed modification takes the feature maps and modified weights $\bar{\mathbf{P}}$ to obtain the attention heat map. The grayed boxes represent standard CNN-based componnets

The conventional Faster R-CNN framework was adapted to enable the generation of network attention maps, which are represented as pseudo-color heat maps, highlighted by red lines in Fig. 2.1. Consider a tensor $\mathbf{X}_{a,b,c}$ as a three-dimensional structure with dimensions a, b, and c. Within the standard Faster R-CNN architecture, specifically in the Region Proposal Network (RPN) segment, a conventional CNN produces feature maps $\mathbf{M}_{512,w,h}$, where the depth is 512, and the other two dimensions corresponds to the image's dimensions (w, h). These maps are subsequently multiplied by a weight vector $\mathbf{P}_{512,24}$, resulting in the creation of anchors $\mathbf{A}_{24,w,h}$.

$$\mathbf{A}_{24,w,h} = \mathbf{P}_{512,24}\mathbf{M}_{512,w,h}. \tag{2.1}$$

The anchors $\mathbf{A}_{24,w,h}$ represent the scores for various scales and aspect ratios of the bounding boxes around objects. Half of these anchors (12) are allocated to background scoring, while the remaining half assess the objects.

The goal of generating heat maps is to identify adjusted weights, $\mathbf{R}_{512,1}$, that enable the creation of a unified heat map $\mathbf{H}_{w,h}$ that does not depend on the object's scale and aspect ratio, utilising the feature maps $\mathbf{M}_{512,w,h}$. This process is described through the following equation:

$$\mathbf{H}_{w,h} = \mathbf{R}_{512,1}\mathbf{M}_{512,w,h}. \tag{2.2}$$

The weights $\mathbf{R}_{512,1}$ are derived from the weights $\mathbf{P}_{512,24}$. that are utilised for anchor generation. This calculation proceeds under the premise that only the anchors pertaining to objects are taken into account, and the dimensionality is further diminished by identifying the highest weight (impact) for a given feature across all scales or aspect ratios. During this procedure, the j-th component of the weights $\mathbf{R}_{512,1}$ is determined using the following equation:

$$\mathbf{R}_{512,1}(j) = \max_{i=1,2,\ldots,12} \mathbf{P}_{512,24}(j,i), \tag{2.3}$$

where $\mathbf{P}_{512,24}(j,i)$ stands for the weight that is multiplied by the j-th feature contributing to the i-th anchor. Consequently, an attention heat map was produced that highlights areas likely

to contain one of the recognised categories using "warmer" colours, whereas "cooler" colours indicate pixels likely associated with the neutral background (Fig. 2.2). In a network that has been adequately trained, these warmer areas accurately represent the object's actual location. It should be noted, however, that elevated temperatures may also be observed in zones where objects, despite being entirely distinct, share similar local characteristics as perceived by the network. This mechanism provides a visual examination of the specific areas the network prioritises, allowing for the identification of objects that are frequently confused, even in the absence of visible false positives (i.e. bounding boxes) in the images. The results of experiments are presented in section 6.3.1.

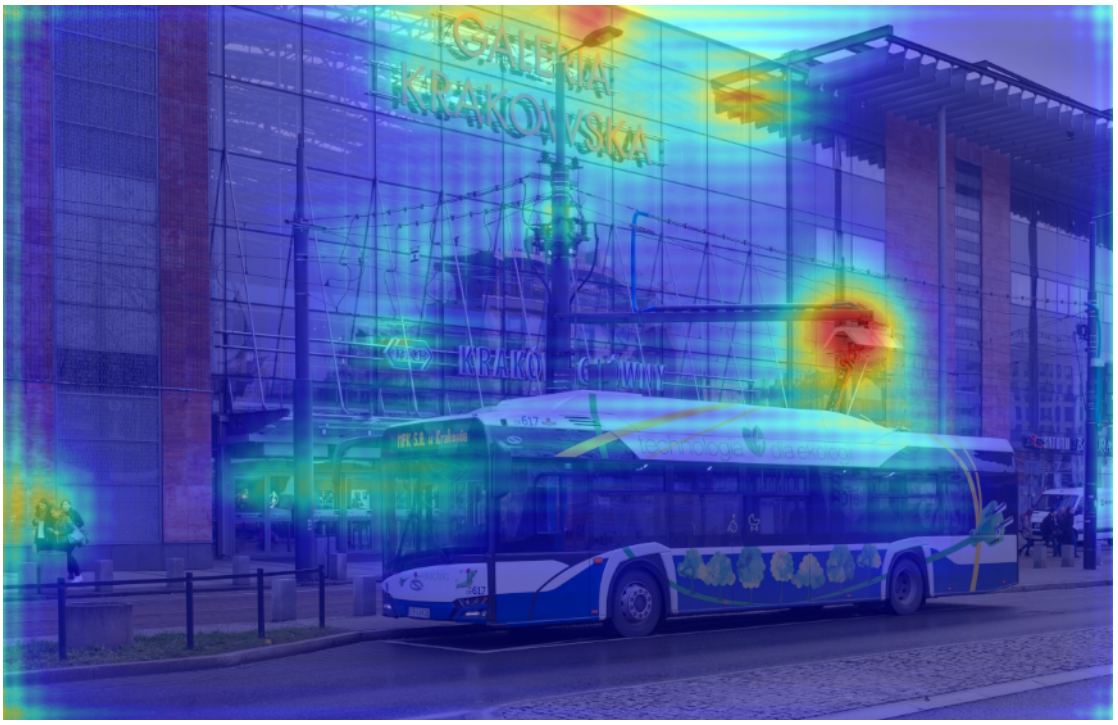

FIGURE 2.2: An example of an attention heatmap. "Warmer" colours highlight areas likely to contain one of the recognised categories, whereas "cooler" colours indicate pixels likely associated with the neutral background

2.3.2 Guided learning

Attention heat maps are an invaluable way to identify and address the factors that are hindering the system's optimal performance. By leveraging this additional insight, the training process can be guided in a more deliberate manner, enriching the dataset with essential examples to enhance system accuracy and efficiency.

The training methodology was developed in several stages, beginning with the creation of a preliminary base model. The initial model underwent training with a dataset that was both limited in size and lacking in diversity, setting the stage for the initial learning phase.

The Faster R-CNN loss function combines a classification loss and a regression loss, represented as follows:

$$\mathcal{L}_{FRCNN} = \frac{1}{N_{cls}} \sum_i \mathcal{L}_{cls}(p_i, p_i^*) + \lambda_{FRCNN} \frac{1}{N_{reg}} \sum_i p_i^* \mathcal{L}_{reg}(t_i, t_i^*), \tag{2.4}$$

where p_i is the predicted probability of the i-th anchor being an object, and p_i^* is the ground-truth label (1 if the anchor is positive, 0 if negative). The classification loss, $\mathcal{L}_{cls}$, is typically the log loss for binary classification:

$$\mathcal{L}_{cls} = -\left(p_i^* \log(p_i) + (1 - p_i^*) \log(1 - p_i)\right) \tag{2.5}$$

The regression loss, L_{reg}, involves the bounding box regression targets t_i and their ground-truth counterparts t_i^* , using a Huber Loss $\mathcal{L}_\delta$:

$$\mathcal{L}_{reg} = \mathcal{L}_\delta(t_i - t_i^*), \tag{2.6}$$

The Huber Loss $\mathcal{L}_\delta$ function is defined as:

$$\mathcal{L}_\delta(x) = \begin{cases} \frac{1}{2}x^2 & \text{if } |x| < 1 \\ \delta \cdot (|x| - \frac{1}{2}\delta) & \text{otherwise,} \end{cases} \tag{2.7}$$

In this formulation, N_{cls} and N_{reg} are normalising factors for the classification and regression losses, respectively. The parameter λ_{FRCNN} balances the relative weight of the regression loss against the classification loss. The Huber Loss δ parameter was equal to 1.

In the subsequent stage of training, the focus shifted to analysing how the network allocated its attention to images sourced from the precise locations where the model was expected to function. The use of images specific to these locations proved to be pivotal, offering a glimpse into the environmental contexts the model would face in real-world scenarios. By visualising the network's focus, valuable insights into the elements within the input images that captured the model's attention was gained. It was also important to identify the elements that the model either ignored or misinterpreted. This process not only revealed the model's current interpretative biases but also identified areas for improvement, guiding further refinements in the training regimen. By adjusting the approach based on these observations, the aim was to rectify misinterpretations and fill in the gaps in the model's understanding, thereby advancing toward a more accurate and reliable object detection system tailored to its intended operational environment.

Utilising the insights obtained from visualising where the network focuses its attention, the causes of the wrong detections were understood. These wrong detections were mostly due to objects having similar lower-level features as the target object, but appearing more often in the training data. As the countermeasure to this problem, the training set was expanded by adding a selection of negative samples, chosen to target and rectify the deficiencies uncovered during the preliminary training phase. These additional samples were sourced from publicly accessible online resources, ensuring the augmentation process remained both cost-effective and efficient.

With the dataset thus enriched, a subsequent round of training was embarked upon. This phase initiated with the foundational base model, which had been established in the initial phase of training, acting as the starting point. This iterative process of refining the initial model through the incorporation of selected negative examples and subjecting it to further training progressively enhances the model's precision and resilience. The Block diagram of the processing pipeline is shown in Fig. 2.3.

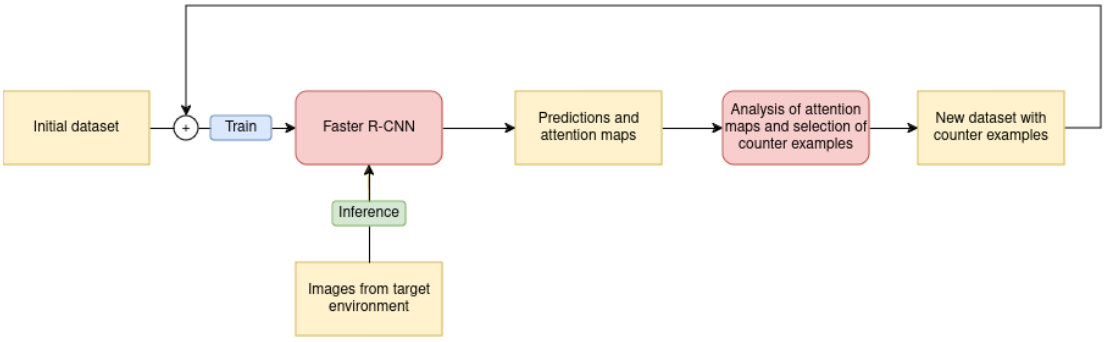

FIGURE 2.3: A block diagram of the pipeline for attention maps based guided learning.

This gradual training strategy, which transitions from an elementary model trained on a constrained dataset to a more sophisticated one refined with data augmented by insights derived from attention visualization, exemplifies an approach to cultivating robust machine learning models under conditions of limited data availability. It embodies the principle of adaptive learning, whereby the model is not only informed by the data it is trained on but also by an ongoing analysis of its performance and by the changing demands of the environment it is designed to operate within. This approach ensures that the model continually evolves and adapts, enhancing its capability to perform its designated tasks with increasing accuracy and reliability. The results of guided learning experiments are presented in section 6.3.2.

Chapter 3

Estimation of 2D characteristic points

3.1 Introduction

One of the fundamental tasks in the field of computer vision is keypoint estimation. This process involves the identification and localisation of specific points of interest within an image. These keypoints are typically defined as salient, easily identifiable locations in the image, such as human body joints [18], corners, or other distinct features [93]. In 2D keypoint estimation, the objective is to determine the coordinates of these points on a two-dimensional image plane. This process is important for understanding the structure and geometry of objects within the image, forming the basis for more complex image analysis and interpretation tasks.

In the context of autonomous driving, 2D keypoint estimation plays an important role as it underpins a variety of applications that are necessary for ensuring the safe and efficient operation of autonomous vehicles. For instance, keypoint estimation is integral to the processes of lane detection, where keypoints on lane markings help maintain the vehicle's trajectory [45]. Similarly, in the context of object tracking and pedestrian detection, the identification of keypoints on moving objects enables the vehicle to monitor and predict their movements [62].

Nowadays, 2D kepoint estimation is primarily solved with neural networks. A significant amount of research focuses on the estimation of keypoints on the human body, known as human pose estimation [125]. CNNs are effective for this task due to their ability to learn hierarchical features and spatial relationships within images. A common technique is heatmap regression, where CNNs predict heatmaps indicating the probability of keypoint locations.

Despite its significance, 2D keypoint estimation faces a number of challenges. One of the primary issues that must be addressed is the loss of spatial resolution that occurs during the image processing stage. As images pass through layers of a convolutional or transformer neural network, the loss of spatial resolution can result in the obscuration or loss of finer details, which are necessary for accurate keypoint detection. This loss of resolution can have a significant impact

on the accuracy of the keypoints detected, potentially leading to errors in subsequent tasks such as pose estimation.

A further challenge is the selection of appropriate, well-defined characteristic points. The efficacy of keypoint estimation is highly dependent on the capacity to identify points that are not only distinctive and stable across disparate views but also relevant to a specific task. In autonomous driving, where environmental conditions and viewpoints can vary significantly, the selection of such characteristic points becomes even more challenging. The keypoints must be robust to changes in lighting, weather conditions, car type, and other external factors, ensuring consistent performance regardless of the external environment.

Another challenge, especially important in autonomous driving, is the estimation of uncertainty associated with the prediction. The essence of this task in keypoint predictions lies in its ability to estimate the covariance matrix of the predicted coordinates. In the dynamic and often unpredictable environments where autonomous vehicles operate, the clarity and accuracy of keypoint detection can be affected by numerous factors, including occlusions, variable lighting, and diverse weather conditions. In such scenarios, the ability to quantify the uncertainty of keypoint predictions empowers the vehicle's decision-making systems to evaluate the reliability of the information being processed. It is not merely a technical necessity but a safety imperative to understand and quantify the uncertainty in these predictions. Furthermore, this feature of uncertainty measurement serves as a feedback mechanism for the continuous learning and adaptation of the system [88]. By identifying areas where uncertainty is significant, a direction for further model refinement and data collection is provided, thus the overall robustness of the perception system is enhanced.

In addition to uncertainty estimation, the interpretability of 2D keypoint predictions represents an important factor. It is of great importance that autonomous driving systems are able to provide an explanation as to why a particular prediction has been made. This is essential for the trust and validation of the system. Interpretability ensures transparency in the decision-making process, thereby facilitating better diagnostics, error analysis, and system improvements. Moreover, it plays an important role in the regulatory and ethical aspects, providing clarity and justification for the vehicle's actions, which is essential for the wider acceptance and trust in autonomous driving technologies [46].

Geometric reasoning uses knowledge of 3D object dimensions, obtained either through predefined models or learned characteristics, to improve the accuracy of estimating 2D keypoints. The application of predefined models is limited to objects with standard dimensions, such as traffic infrastructure elements such as signs and lights. In the majority of cases, the objects in question possess custom shapes or the number of possible 3D models is exceedingly large, such as in the case of different vehicle models. In such instances, the approach that has gained increasing interest in recent years is to utilise a neural network-based reconstruction of the 3D model of the considered object from images [109], [111].

Geometric reasoning plays a significant role in enhancing the accuracy and reliability of 2D keypoint estimation. These approaches facilitate a more in-depth understanding of the spatial relationships and physical dimensions of objects within the environment, which are important for

the safe navigation of vehicles. One of the primary benefits of geometric reasoning is its capacity to provide context to the keypoints. By comprehending the geometric interrelationships between disparate keypoints, the system is better positioned to anticipate their positions in disparate scenarios, thereby enhancing the resilience of the detection process [38].

In this research, the problem of 2D semantic keypoints estimation from monocular camera images is explored under two specific scenarios. The first focuses on a docking maneuver involving an electric city bus approaching a charging station, where the geometry of the charging station is predefined and the viewpoints from which the bus approaches are restricted, providing a relatively controlled setting for the estimation process. The challenge here involves the precise localisation of keypoints on the charging station to guide the docking maneuver effectively.

The second scenario considers the estimation of the pose of surrounding vehicles in an urban environment. This situation presents a greater level of complexity due to the diverse shapes and geometries of vehicles. Moreover, an environment is dynamically changing with the common presence of occlusions making this less predictable setting.

The keypoints that are estimated are used to compute the pose of an object relative to the camera using an algorithm solving Perspective-n-Point problem. This step is integral for making navigation decisions in autonomous driving applications, such as trajectory adjustment or precise alignment with a charging station. The challenge extends beyond accurate keypoint detection to ensuring that these keypoints are reliably positioned for effective pose estimation across a variety of operational contexts.

This chapter presents the following contributions to the field of 2D keypoint estimation:

- New neural network architectures designed for estimation of keypoints on the road infrastructure objects
- A new loss function that leverages knowledge about the 3D object model applied in the vehicle points estimation
- A postprocessing procedure that refines keypoint predictions based on the 3D object model

3.2 Related work

The topic of keypoint detection is frequently discussed in the literature, with main applications in human pose estimation. The developments in this field can be divided into two predominant approaches: regression-based and heatmap-based methods. Each approach offers distinctive advantages and has been adapted to address specific challenges in keypoint detection tasks. Regression-based keypoint detection entails the direct prediction of the numerical coordinates of keypoints from images [97]. This method simplifies the processing pipeline by eliminating intermediate steps such as heatmap generation, thus potentially increasing computational efficiency. Regression-based methods for human pose estimation have historically been less accurate than heatmap-based approaches due to their inability to effectively utilise structural pose information. In [94], Sun described an approach that utilises a reparameterised pose representation that

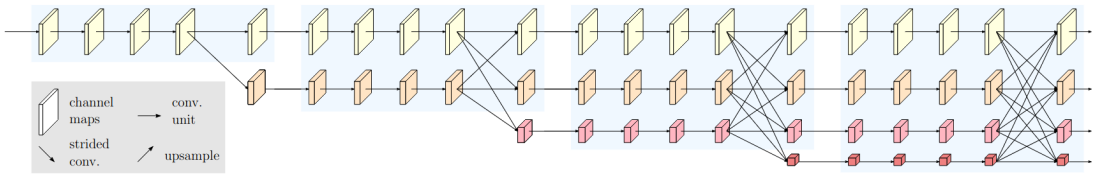

FIGURE 3.1: Architecture of the HRNet. Source: [107]

focuses on bones rather than joints. The approach leverages the structure of joint connections to establish a compositional loss function, which captures the long-range interactions within the pose. The regression-based approach described in [63] employs a Transformer network to directly regress keypoint coordinates from image data. This method incorporates an attention mechanism to improve feature alignment and is the first regression-based model to rival the performance of heatmap-based approaches, as demonstrated on datasets such as MS-COCO [57] and MPII [2].

Conversely, heatmap-based approaches result in the generation of spatial heatmaps, wherein each pixel's value represents the probability of a keypoint's presence at that location. Both regression-based and heatmap-based methods can be further divided into two primary methodologies: top-down and bottom-up. The bottom-up approach initially identifies all the body parts in the image, regardless of the individual, and subsequently assembles them into distinct human poses. This approach offers a more efficient processing flow, particularly when scaling to a larger number of people. The Part Affinity Fields (PAFs) introduced in [13] are a unique approach that associates body parts with individuals in an image while encoding global context. This allows for real-time performance through a bottom-up parsing step that remains highly accurate regardless of the number of people present.

The top-down approach first detects each individual in an image and then predicts their pose. This effectively handles each person separately, which can be advantageous in crowded scenes but is computationally expensive. The HRNet, as described in [107], is designed to maintain high-resolution representations throughout the networks processing stages. The architecture begins with a high-resolution subnetwork and gradually incorporates high-to-low resolution subnetworks. Parallel connections and repeated multi-scale fusions enhance the richness of the high-resolution output (Fig. 3.1). Newell [74] presents a method that fuses top-down and bottom-up approaches, a new convolutional network architecture designed for human pose estimation, termed the "stacked hourglass" network. This architecture effectively processes features at all scales and integrates them to capture the complex spatial relationships of body parts. By employing a strategy of repeated bottom-up and top-down processing with intermediate supervision, the network significantly enhances performance.

The literature addressing the topic of keypoint detection for vehicles is relatively limited. The paper [95] evaluates the effectiveness of simple baseline methods by incorporating deconvolutional layers into a backbone network to generate heatmaps for vehicle keypoints - a technique that has already been successfully applied to human pose estimation. The results, validated on the PASCAL3D+ dataset [116], achieved state-of-the-art results. Furthermore, additional experiments highlighted existing issues in vehicle keypoints labelling. A novel approach to defining

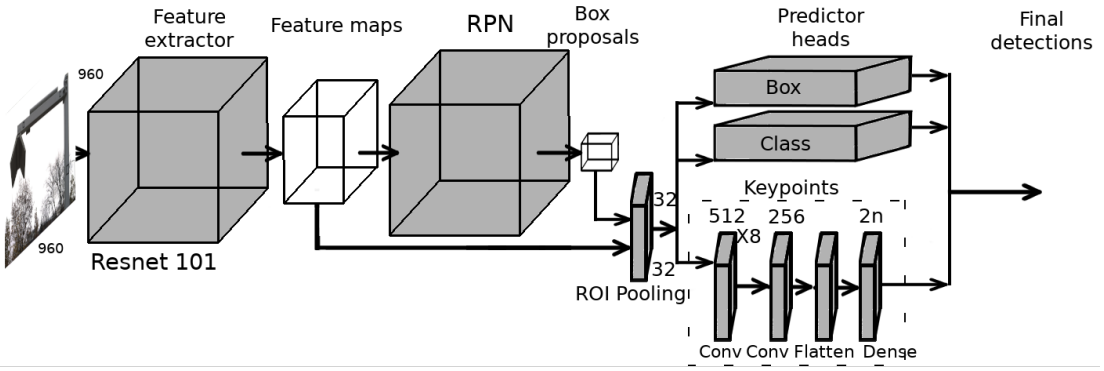

FIGURE 3.2: Block diagram of the Regression Keypoint Network (RKN) architecture for the keypoint detection.

vehicle keypoints was introduced which was validated using a customised dataset with extended keypoints, which addressed the aforementioned performance challenges. In [47], Kreiss presented a comprehensive framework that simultaneously detects and establishes spatiotemporal keypoint connections in a single step, marking the inception of the first real-time pose detection and tracking algorithm. This approach has been shown to be applicable to various semantic keypoints, including those pertaining to cars.

The reprojection loss has previously been employed for the estimation of human 3D poses. In [37], Kanazawa et al. employed reprojection loss for the purpose of 3D human mesh recovery. This permitted the model to be trained using images captured in the wild, which only had ground truth 2D annotation. Similarly, in [82], reprojection loss was employed for the self-supervised training of the network.

3.3 Proposed network architectures

Both heatmap-based and regression-based keypoint estimation approaches offer unique advantages and face distinct challenges. The choice between these methods depends on the specific requirements of the application, including the need for precision, computational efficiency, and the ability to handle spatial ambiguity. As the field of computer vision advances, ongoing research continues to explore these methodologies, seeking to leverage their strengths and mitigate their weaknesses for improved 2D keypoint estimation.

3.3.1 Keypoints detection: Regression Keypoint Network

This section will present a regression-based neural network for the detection of keypoints on the electric bus charger. The well-known Faster R-CNN network serves as the foundation for the development of the Regression Keypoint Network (RKN), an object detection framework inspired by the former's architecture. The initial stage of this process includes the transformation of the input image by a backbone network, which is responsible for deriving a series of feature maps from the image. Specifically, the architecture that has been adopted (illustrated in Fig. 3.2) employs the ResNet101 as the backbone, which generates a set of 1024 feature maps. Subsequently,

the aforementioned maps are forwarded to the Region Proposal Network (RPN), a component designed to identify and propose a collection of regions within the image that are highly likely to encapsulate objects belonging to the targeted category.

Subsequently, the next step involves the extraction of fragments from the backbone's feature maps, following the identification of the aforementioned regions by the RPN. Further, the selected regions are subjected to standardisation through the application of the ROI Pooling layer, thereby ensuring uniformity in their dimensions. This uniformity is important, as it allows these regions to be processed in parallel by the various heads of the predictor, thus enhancing the overall efficiency of the system. In order to enhance the resolution of the processed regions and to minimise the loss of spatial information, adjustments were made to the parameters of the RPN network. The objective was to retain as much spatial detail as possible within these regions.

Finally, the regions delineated by the RPN are resized to a consistent dimension of 32×32 pixels. This dimension represents the upper limit of what can be accommodated within the memory constraints of the available GPU, striking a balance between maintaining sufficient detail for accurate keypoint detection and adhering to the hardware limitations. This process serves to illustrate the network's design philosophy, which prioritises the preservation of spatial information that is critical for the accurate localisation of keypoints within the constraints of the computational resources. It is assumed that the bottleneck limiting the accuracy of keypoint localisation in this approach is the resizing of regions of interest to a 32×32 px size.

The first head of the RKN architecture is a regressor tasked with refining the position of the bounding boxes. The second head is designed to ascertain the class association and the confidence level of each proposed region. In contrast to the original Faster R-CNN design, the architecture incorporates a third predictor, which is tasked with determining the positions of keypoints within the images.

The keypoint prediction module is constructed from a sequence of eight convolutional layers, each equipped with 512 filters. Subsequently, the architecture incorporates an additional convolutional layer, comprising 256 filters. Each convolutional layer uses a filter size of (3,3). The output of the final convolutional layer is then flattened into a vector. Subsequently, the vector undergoes further processing by a densely connected layer, which is specifically designed to produce a number of values that is twice the amount of the predefined keypoints. The final outputs are tailored to represent the precise x and y coordinates for each keypoint directly, offering an accurate estimation of their positions.

The loss function used during training is the same as in (2.4), extended with $L_{MSEkpts}$ defined by Eq. (3.1):

$$\mathcal{L}_{MSEkpts} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left\|\mathbf{p}_i^{2d} - \hat{\mathbf{p}}_i^{2d}\right\|^2}, \tag{3.1}$$

where $\mathbf{p}_i^{2d}$ is the ground truth 2D point, $\hat{\mathbf{p}}_i^{2d}$ is the predicted 2D point and n denotes the total number of points.

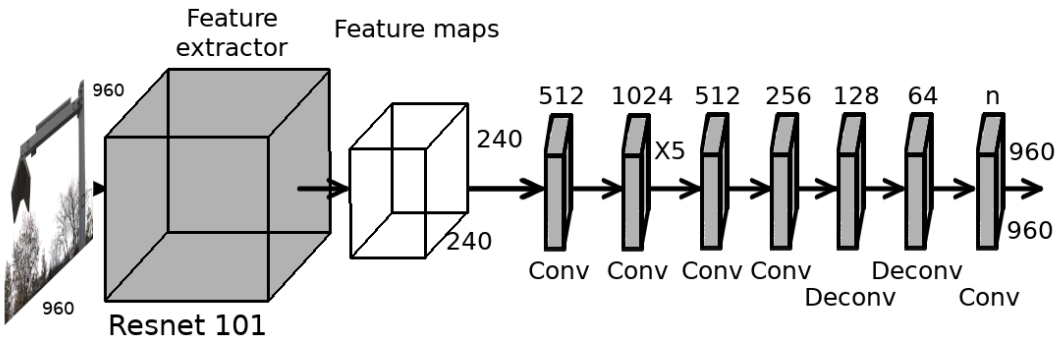

FIGURE 3.3: Block diagram of the Max Resolution Heatmap Keypoint Network (MRHKN) architecture for the keypoint detection.

The final loss function is defined by Eq. (3.2).

$$\mathcal{L}_{RKN} = \frac{1}{N_{cls}} \sum_i \mathcal{L}_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* \mathcal{L}_{reg}(t_i, t_i^*) + \mathcal{L}_{MSEkpts} \tag{3.2}$$

3.3.2 Keypoint detection: Max Resolution Heatmap Keypoint Network

The alternative method for keypoint localisation was conceived as a result of the realisation that the conventional heads within the RKN architecture could be considered redundant, given that it is assumed that the object of interest has already been detected and its bounding box coordinates are known. Furthermore, it was observed that the bottleneck of the size 32×32 px in RKN network impedes the subsequent head's ability to estimate keypoints at a higher resolution. In light of these observations, an innovative architectural concept was introduced, named the Max Resolution Heatmap Keypoint Network (MRHKN), specifically crafted to preserve the highest possible resolution of the input image, as depicted in Fig. 3.3.

This architecture represents a departure from the RKN framework, as it eliminates the RPN along with all other heads, except for the one dedicated to keypoint detection. This omission allows for a notable expansion in both the depth and width of the keypoint detection head. The streamlined architecture of the MRHKN model is designed to optimise the utilisation of the input image's resolution. This focus on maintaining image resolution throughout the processing pipeline is an important aspect of the new design, which aims to enhance the keypoint detection capability.

In a similar manner to the methodology employed in the RKN configuration, the architectural design utilises the ResNet101 network as its backbone. This block generates the feature maps, which are subsequently reduced in size by a factor of four. Consequently, for an input image measuring 960×960 pixels, this reduction process yields feature maps with a resolution of 240×240

pixels. Subsequently, these feature maps undergo a series of transformations through a sequence of eight convolutional layers, without further noticeable resolution loss.

In order to ensure that the output heatmap and the original input image shapes are identical, two deconvolution layers have been incorporated into the workflow. These layers play a pivotal role in gradually enlarging the feature maps to their original dimensions, thereby facilitating a direct correlation between the heatmap and the input image dimensions. The final stage of the process involves the application of a final convolutional layer, which is distinguished by the utilisation of filters with a size of (1,1). The primary function of this layer is to craft a distinct heatmap for each of the n keypoints under consideration.

The produced heatmap provides only an indication of the likelihood of positions for the key points. Consequently, additional steps are required to extract precise coordinates for these points. An illustration of the post-processing required is shown in Fig. 3.4. In Fig. 3.4A, an actual keypoint location is highlighted by the red circle. The initial output from the network, as depicted in Fig. 3.4B, also presents false positive detections, which should be removed. These are indicated by red arrows. A detailed view of an accurate positive detection is presented in Fig. 3.4C, while Figs. 3.4D and E showcase examples of false activations. To eliminate these incorrect activations, a process of thresholding is employed to convert the heatmap into a binary format (shown in Fig. 3.4F). Subsequently, the DBSCAN clustering algorithm is applied to the binary image to identify potential keypoint locations. The selection of the most probable keypoint cluster is based on the calculation of a confidence score for each potential cluster. In particular, the confidence score S_i for a given cluster K_i is calculated by summing the intensity values $I(\mathbf{x})$ for each pixel location $\mathbf{x} = [u, v]$ within the raw heatmap that is part of cluster K_i.

$$S_i = \sum_{\mathbf{x} \in K_i} I(\mathbf{x}) \tag{3.3}$$

The final keypoint location $\mathbf{c}_i$ is computed as the center of mass of the cluster with the highest confidence score (Fig. 3.4G):

$$\mathbf{c}_i = \frac{1}{S_i} \sum_{\mathbf{x} \in K_i} \mathbf{x} \cdot I(\mathbf{x}) \tag{3.4}$$

For training the MRHKN network, a binary cross-entropy loss was used:

$$\mathcal{L}_{BCE} = \sum_{i=1}^{n} \left(-\frac{1}{m} \sum_{j=1}^{m} [m_{ij} \log(\hat{m_{ij}}) + (1 - m_{ij}) \log(1 - \hat{m_{ij}})] \right), \tag{3.5}$$

where m_{ij} is the true label for the j-th pixel of heatmap corresponding to i-th point, and $\hat{m_{ij}}$ is the predicted value of considered pixel. In this equation, m denotes the total number of pixels in the predicted heatmap, and n is the number of points.

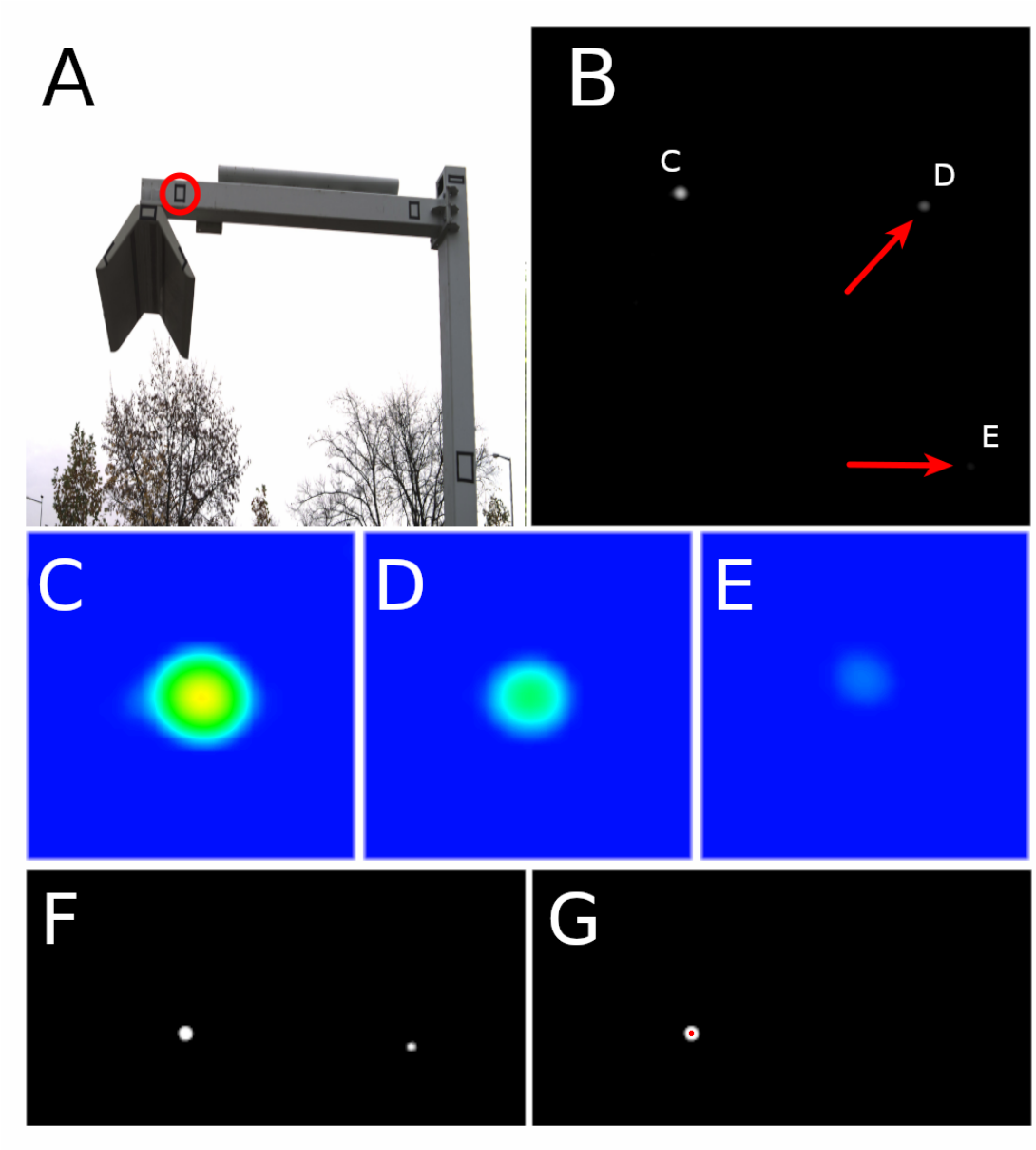

FIGURE 3.4: MRHKN postprocessing: input image (A), network output (B), closeup of the network output near a marker to be found (C), false positive markers (D, E), thresholding (F), DBSCAN clustering and center of mass as actual keypoint coordinates (G)

3.3.3 Keypoint detection: Geometry-Aware Keypoint Network

The genesis of the Geometry-Aware Keypoint Network (GAKN) architecture was significantly influenced by the superior capabilities demonstrated by state-of-the-art keypoint detectors employed in top-down approaches to human pose estimation methods, as referenced in [99]. In the domain of top-down human pose estimation, the strategy includes accurately determining the positions of keypoints within specified bounding boxes that have been outlined by a person detection mechanism.

In the experimental setup, a keypoint detection system that utilises the HRNet architecture [107] as its backbone was employed, which is complemented by a keypoint detection head, implemented

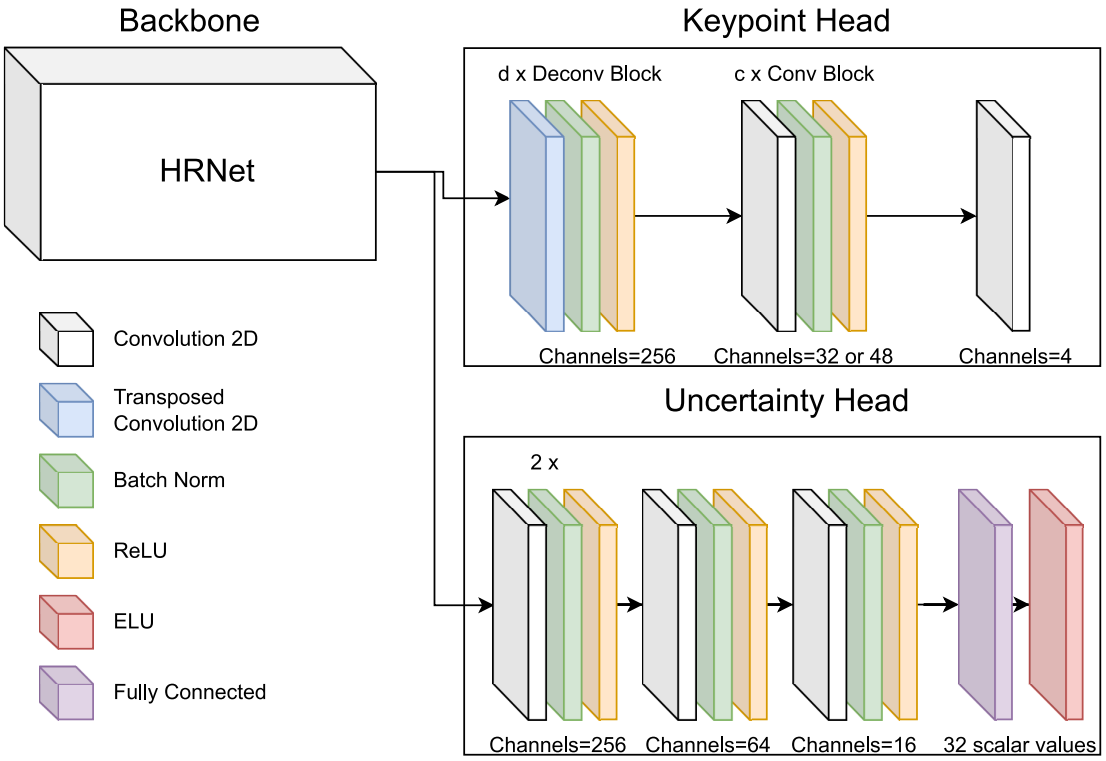

FIGURE 3.5: Architecture of the GAKN model. Configurations with heatmap sizes of 128×128, 256×256 and 512×512 contain d equal 0, 1 or 2 Deconv Blocks respectively. Architectures can be extended with c additional Conv Blocks in the keypoint head

within the MMPose framework [67]. The network heads were designed specifically for this task, including an additional head aimed at assessing the spatial uncertainty associated with the detected keypoints, as shown in Fig. 3.5.

The detection head is constructed using deconvolutional blocks, which have been designed to enhance the resolution of the feature maps by a factor of two. Each deconvolutional block comprises a series of layers, including a transposed convolution layer, a batch normalization layer, and a Rectified Linear Unit (ReLU) activation layer. At the end of the detection head is a single convolutional layer, which is responsible for producing a set of heatmaps, which are equal in number to the keypoints being targeted.

In order to generate precise training targets in the form of ground truth heatmaps and to accurately interpret the final keypoint positions from these heatmaps, the Unbiased Data Processing (UDP) and DarkPose techniques were integrated, as outlined in [35] and [122], respectively. These preprocessing methods play an important role in ensuring the integrity of the keypoint coordinates throughout the preprocessing and augmentation phases. Moreover, when it comes to interpreting the predicted heatmaps during the inference stage, these methodologies facilitate the extraction of keypoint locations with remarkable precision, down to subpixel accuracy.

In this study, an assessment of two distinct backbone networks, HRNet32 and HRNet48 was conducted. The primary distinction between HRNet48 and HRNet32 is found in the width of the convolutional layers within the high-resolution pipeline, which are 48 and 32 channels wide,

respectively. This change gives HRNet48 a better ability to create more feature maps, but it also requires more computational resources.

Furthermore, the effect of incorporating additional convolutional layers into the keypoint detection head was explored to ascertain their impact on enhancing the accuracy of pose estimation. These supplementary layers were positioned subsequent to the Deconvolutional Block and prior to the convolutional layer responsible for generating the ultimate heatmaps. Each of these added layers is equipped with 256 filters measuring 3×3, followed sequentially by Batch Normalization and ReLU activation functions.

It is noteworthy that the standard configuration of the keypoint detector utilising HRNet outputs heatmaps that are downscaled by a factor of four relative to the original size of the input image. Consequently, an input image with dimensions of 512×512 pixels yields heatmaps with dimensions of 128×128 pixels. The process of upsampling these heatmaps is facilitated through the application of Transposed Convolutional layers, with a single layer effectively doubling the width and height of the heatmap.

Given that the components of the charging station, including the pylon and head, are static and maintain a consistent geometric arrangement, the preliminary localisation of keypoints is somewhat more straightforward compared to the dynamic and complex nature of human pose estimation. This static nature allows for the anticipation of keypoints within specific regions of the image. However, the precision of this pose estimation technique is heavily reliant on the accuracy with which keypoints are identified. Even minor deviations in the pinpointing of keypoints can lead to disproportionately significant errors in pose estimation, particularly when observed from extended distances. These observations lead to the hypothesis that enhancing the resolution of the output heatmaps could significantly contribute to the precise subpixel determination of keypoint positions, thereby markedly improving the accuracy of camera pose estimation.

It is important to highlight the difference between detecting keypoints on objects such as an electric bus charger and detecting keypoints on the human body. The human pose consists of many parts that move independently, making it less straightforward to determine which part of an image to analyse for a specific keypoint. However, the fixed geometric structure of a charging station simplifies this aspect in the network's design, allowing us to concentrate the efforts on refining the precision of keypoint localisation.

The importance of scene geometry within traditional models dedicated to estimating poses, combined with the straightforward process of identification points for elements like the charging station, as well as understanding the spatial relationships among these specific points, served as the catalyst for evolving the HRNet baseline model. This evolution led to the creation of the Geometry-Aware Keypoint Network. The concept was driven by the realization that a deep comprehension of the geometric structure of a scene significantly enhances the model's ability to accurately predict poses. By integrating these geometric principles directly into the model's architecture, the aim was to leverage the natural structure and layout of the scene, particularly the charging station, to inform and refine the network's predictions. This approach represents a deliberate shift towards a more geometry-focused methodology in keypoint detection and pose

estimation, aiming to harness the inherent spatial cues present within the environment to improve the accuracy and reliability of the model's outputs.

During its training phase, this network leverages geometric priors by incorporating an extra loss function that is grounded in the concept of reprojection error. This reprojection loss is designed to impose penalties on any spatial arrangements of keypoints that defy the constraints of physical reality, thereby discouraging the model from predicting keypoint configurations that cannot exist in the physical world. The camera projection function, π, is defined which maps the i-th 3-D point $\mathbf{p}_i^{3d}$ to the 2-D image point $\tilde{\mathbf{p}}_i^{2d}$ leveraging the given camera intrinsics parameters $\mathbf{K}$:

$$\pi(\mathbf{T}, \mathbf{K}, \mathbf{p}_i^{3d}) \mapsto \tilde{\mathbf{p}}^{2d}, \tag{3.6}$$

where $\mathbf{T}$ is a rigid transformation matrix (rotation and translation).

To calculate the reprojection loss, the difference between the projection of the real 3-D $\tilde{\mathbf{p}}^{2d}$ object points is minimised and the points predicted by the network $\hat{\mathbf{p}}^{2d}$. The optimisation problem is solved by the Trust Region Reflective algorithm [10] which finds a transformation $\mathbf{T}^*$:

$$\mathbf{T}^* = \underset{\mathbf{T}}{\operatorname{argmin}} \sum_{i=1}^{n} \|(\tilde{\mathbf{p}}_i^{2d} - \hat{\mathbf{p}}_i^{2d}|\|_1^2. \tag{3.7}$$

The Trust Region Reflective algorithm stands out as an optimization method designed to tackle constrained problems. In this application, to ensure that the solution remains within the bounds of physical feasibility and accurately represents the real-world scenario, specific constraints on the transformation matrix $\mathbf{T}$ were introduced. These constraints are designed to mirror the actual operational environment of the bus. For instance, the range of all three components of the rotation vector was restricted to $\frac{\pi}{4}$. This limitation is based on the assumption that the roll and pitch angles of the bus are negligible, thereby confining the yaw angle variation to within $\pm\frac{\pi}{4}$, ensuring the rotation remains realistic and within the expected operational parameters.

Furthermore, the constraints extend to the translation movements of the bus, with the lateral (side-to-side) movement restricted to within $\pm$ 20 meters to reflect the typical maneuvering space. The forward or backward movement, represented by the longitudinal axis, is constrained to 50 meters, acknowledging the usual operational range. Additionally, the vertical translation, or movement along the z axis, is also capped at 50 meters, providing a comprehensive framework that effectively delineates the search space for the optimisation process, as illustrated in Fig. 3.6. These constraints are required for guiding the optimisation towards solutions that are not only mathematically sound but also align with the physical constraints and operational realities of bus movement, ensuring the model's predictions are grounded in the physical world. For the initial guess for optimisation, a point within the bus operational space defined by the aforementioned constraints was chosen.

Finally, the reprojection loss is the value of the cost function for the optimal $\mathbf{T}^*$ transformation:

$$\mathcal{L}_{reprojection} = \sum_{i=1}^{n} \|(\pi(\mathbf{T}^*, \mathbf{K}, \mathbf{p}_i^{3d}) - \hat{\mathbf{p}}_i^{2d}\|_1^2. \tag{3.8}$$

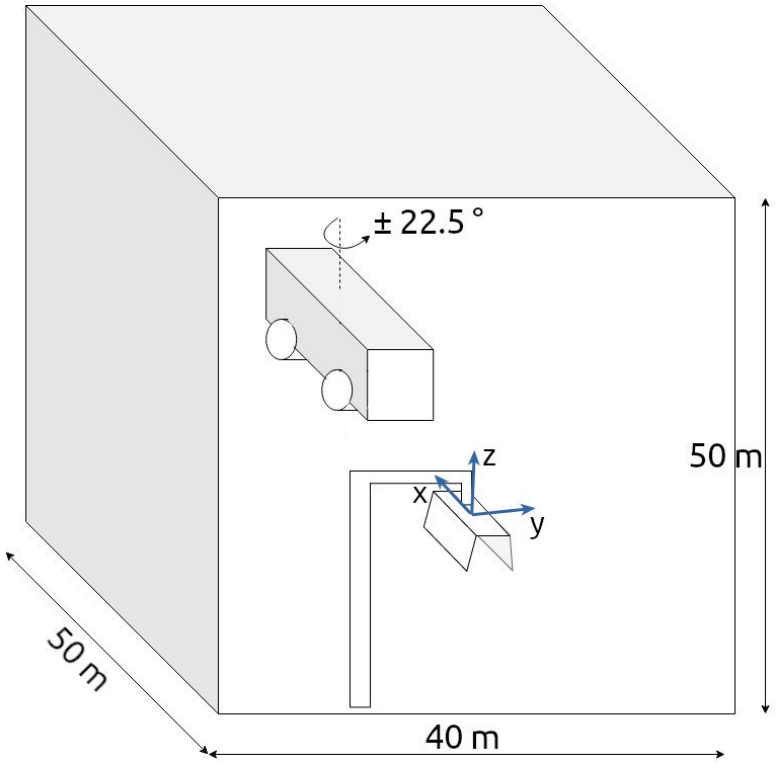

FIGURE 3.6: The considered search space for the bus location and orientation.

The second loss element is a Mean Squared Error Loss:

$$\mathcal{L}_{MSE} = \sum_{i=1}^{n} \left(\frac{1}{m} \sum_{j=1}^{m} (m_{ij} - \hat{m_{ij}})^2 \right), \tag{3.9}$$

where m_{ij} is the j-th pixel of the ground truth heatmap corresponding to the i-th point and $\hat{m_{ij}}$ is the j-th pixel of the predicted heatmap corresponding to the i-th point. The final loss function is a sum of the Mean Squared Error (MSE) loss (3.9) and the reprojection loss, calculated according to the formula:

$$\text{loss} = \mathcal{L}_{MSE} + \lambda_{reprojection} \cdot \mathcal{L}_{reprojection}, \tag{3.10}$$

where the metaparameter $\lambda_{reprojection}$ was estimated experimentally and is set to 0.01 for the evaluation. The experimental evaluation of the aforementioned networks is presented in section 6.4.

3.4 Uncertainty estimation of 2D points coordinates

In this section, the discussion delves into a module designed to estimate the uncertainty associated with the estimation of characteristic points.

A covariance matrix of size $2n \times 2n$, denoted as $\mathbf{\Sigma}_{2D}$, is estimated to represent the uncertainties of n points in the x and y dimension. The Uncertainty Estimation Head (UEH) takes the image feature maps as input. It processes these inputs to calculate $(2n + 1)n$ positive numbers, which

populate the lower triangle of the 2D Cholesky factor matrix, $\mathbf{L}_{2D}$. The UEH is comprised of a stack of four blocks, each constructed from a convolutional layer, a ReLU activation function, and batch normalization. The numbers of output filters for these convolutional layers are: [256,256,64,16]. Following these convolutional blocks is a single linear layer that outputs $(2n+1)n$ values. The architecture of UEH is shown in Fig. 3.5. To ensure that the estimated numbers are positive, an Exponential Linear Unit (ELU) activation function is applied, followed by the addition of a constant to the output values. By multiplying the matrix $\mathbf{L}_{2D}$ by its transpose $\mathbf{L}_{2D}^T$, the covariance matrix $\mathbf{\Sigma}_{2D}$ is derived. This methodology guarantees that the resulting covariance matrix is positive semi-definite which is a critical property for a valid covariance matrix.

During the training phase, the Gaussian Log-Likelihood Loss function was employed (Eq. (3.11)), as suggested in [49].

$$\mathcal{L}_{unc} = \sum_{i=1}^{n} \log |\mathbf{\Sigma}| + \left(\mathbf{p}_i^{2d} - \hat{\mathbf{p}}_i^{2d}\right) \mathbf{\Sigma}^{-1} \left(\mathbf{p}_i^{2d} - \hat{\mathbf{p}}_i^{2d}\right), \tag{3.11}$$

where $\mathbf{p}_i^{2d}$ is a vector of ground truth keypoint locations and $\hat{\mathbf{p}}_i^{2d}$ is a vector of predicted keypoint locations.

The 2×2 covariance matrices of the individual keypoints are extracted from $\mathbf{\Sigma}$ and can be visualised as uncertainty ellipses in the image plane. The evaluation of this uncertainty estimation approach is presented in section 6.7

3.5 Gradient visualization

It was hypothesized that incorporating prior geometric knowledge through a reprojection-based loss component into the Geometry-Aware Keypoint Network architecture would introduce a beneficial inductive bias. This bias was expected to guide the network towards prioritising the most pertinent areas within an image for keypoint identification. To test this theory, the GAKN was augmented with an attention analysis layer, utilising the Score-CAM method [106] for this purpose. In contrast to previous gradient-based approaches, Score-CAM offers a visualization of the network's focus that is clearer and less cluttered by noise, thereby simplifying the task of interpreting how the network's attention varies in response to different input images.

Illustrations in Fig. 3.7 showcase activation maps for four identified keypoints in sample images from the test collection. The top sequence of images for each sample displays heatmaps, revealing that, within the GAKN framework, the activation around individual points appears more focused and exhibits a greater intensity compared to earlier methods. The lower sequence in each example highlights the image segments identified by the Score-CAM method as having the most significant influence on the determination of specific points within the image.

A notable observation from the baseline network's activation maps is the widespread dispersion of activation across the background, suggesting a lack of focused attention by the network. Particularly for point three (third column in each sample), there is minimal activation in proximity to the point's actual location, contrasted with substantial activation in an area of the image not

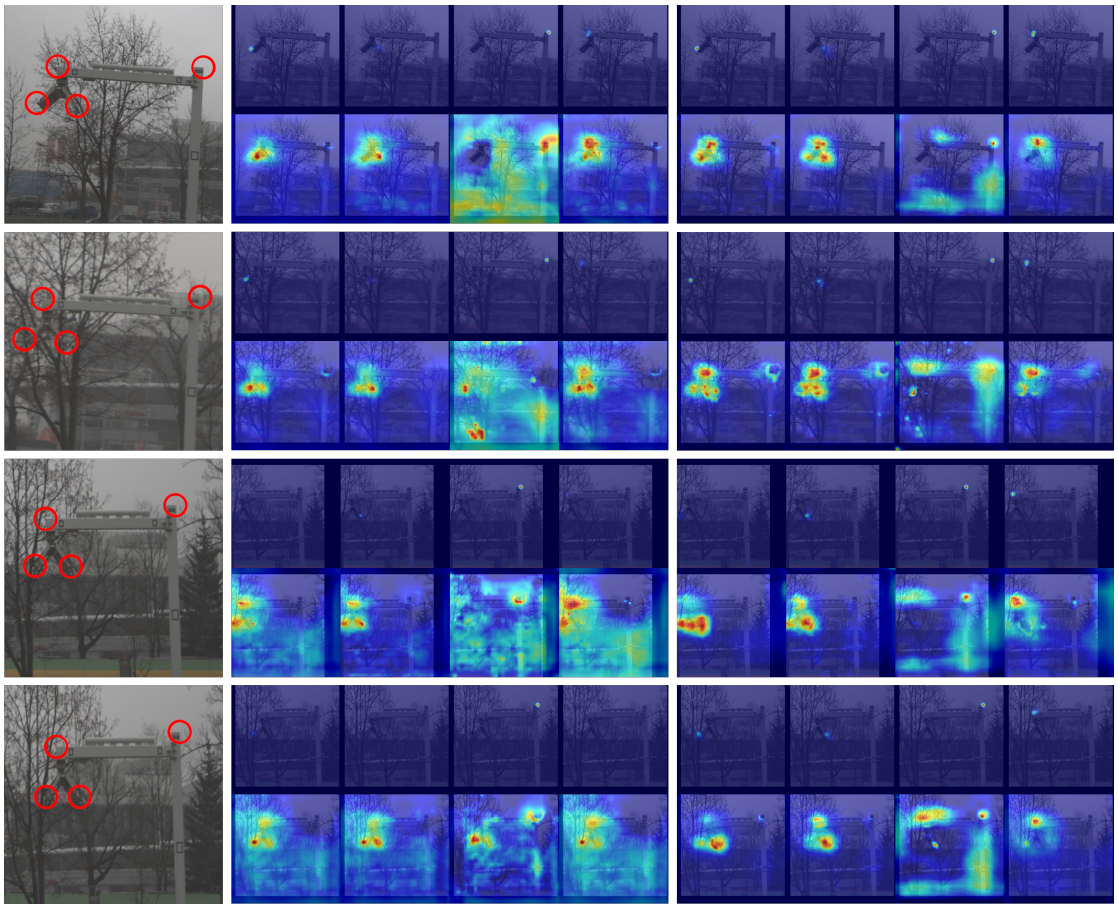

FIGURE 3.7: Visualizations of the outputs of the Score-CAM algorithm and the keypoint detection networks. The first column presents input images with keypoints marked with a red circle. In the second column, there are outputs from the baseline HRNet32 network (top rows) and outputs of the Score-CAM algorithm (bottom rows). The third column presents outputs from the Geometric Aware Keypoint Network (top rows) and outputs of the Score-CAM algorithm (bottom rows). Warmer colors mean higher activation

related to the structure of the charger, specifically in the lower left corner. Conversely, Fig. 3.7 presents activation points for the GAKN network, where it is evident that the network's attention is more narrowly concentrated around the keypoints of interest, distinguishing it markedly from the baseline. This contrast underscores the effectiveness of integrating geometric priors via the GAKN architecture, affirming the initial conjecture that such an approach enhances the network's ability to accurately focus on relevant sections of the image for keypoint detection.

Chapter 4

Estimation of the object's 3D shape

4.1 Introduction

The problem of monocular 3D shape estimation is an intricate challenge within the field of computer vision, also significant in applications of autonomous driving. This task entails the prediction of three-dimensional positions of specific keypoints on an object or a dense mesh from a single two-dimensional image. Unlike methods that rely on multiple images or depth sensors to measure depth, monocular 3D shape estimation must infer depth information from a solitary image, leveraging cues such as perspective, shading, and context.

The main challenge in monocular shape 3D estimation arises from the inherent loss of depth information when capturing a three-dimensional scene in a two-dimensional image. This transformation, while preserving information about the lateral and vertical axes, results in an ambiguity regarding the z-axis (depth) for any given point in the image. As a consequence, the task demands sophisticated inference techniques that can accurately reconstruct the lost depth information based on monocular cues. This process involves a complex understanding of geometric principles, object appearance, and scene context to predict the likely 3D structure that corresponds to the two-dimensional visual input [24].

Despite the considerable challenges, advancements in machine learning particularly in deep learning have led to significant progress in this area. Neural networks have been at the forefront of these advancements, demonstrating the ability to learn powerful feature representations from data that encapsulate the complex mappings required for accurate 3D shape estimation from monocular images. These networks are trained on large datasets containing images annotated with the 3D positions of keypoints, learning to generalise from these examples to new, unseen images [24].

Another significant challenge encountered in the processing of images by neural networks is the loss of spatial resolution, a phenomenon that invariably leads to a decrease in the accuracy

of the results. As images pass through the successive convolutional and pooling layers of a CNN, detailed spatial information tends to get less distinct. This is primarily because these layers are designed to abstract and compress the image data to extract relevant features for the task at hand, such as object detection or classification. However, this process of abstraction and compression often results in the loss of fine details and nuances of the spatial arrangement within the image. For tasks that require high fidelity in spatial localization, such as precise keypoint detection or segmentation, the diminished spatial resolution can significantly hamper the network's performance. The challenge, therefore, lies in designing CNN architectures and processing strategies that can maintain or recover spatial resolution to ensure high accuracy of the results, especially in applications where precise spatial details are critical for success.

However, the problem is not solely a matter of neural network architectures but also involves considerations related to the availability of high-quality annotated training data, computational efficiency, and their robustness to variations in lighting, occlusion, and background. Moreover, the practical deployment of monocular 3D shape estimation systems necessitates their integration into broader systems that can utilise the 3D shape information for tasks such as navigation, object manipulation, or interaction in both real and virtual environments. Such application requires that the estimations from CNN for 3D shape estimation will be explainable and interpretable.

This chapter addresses the problem of estimating 3D geometric features from images captured by a monocular camera. The aim is to determine the 3D coordinates of vehicle keypoints or overall car shapes, which are required for assessing the pose of surrounding vehicles in an urban setting. The methodology relies solely on a monocular camera setup, processing each image independently without the benefit of temporal information. This approach presents challenges, particularly in accurately deriving depth and 3D structure from single images due to the limited depth cues available from one viewpoint. A notable challenge in advancing this research is the scarcity of datasets that provide a direct correlation between semantic 2D keypoints and their corresponding 3D coordinates. This lack of data complicates the development and validation of models that are capable of converting 2D visual information into 3D data.

This chapter outlines the contributions to the field of 3D car shape estimation:

- A new neural network architecture capable of precise estimation of car characteristic points coordinates in 3D space from a single image
- An automatic procedure for obtaining labels of 3D cars' characteristic points in 3D space
- A new neural network architecture for estimation of a dense 3D mesh of cars from a single image

4.2 Related work

The existing approaches to 3D shape estimation can be broadly categorised into two main groups. The first group focuses on predicting a sparse set of 3D characteristic points. These methods typically involve identifying keypoints or landmarks on the object, which provide a simplified but

informative representation of its 3D structure. These sparse keypoint methods are particularly advantageous due to their efficiency and lower computational requirements, making them suitable for real-time applications such as autonomous driving and augmented reality. Li et al. [55] used a simple cuboid representation of cars to estimate 3D pose. This approach was proposed, among other reasons, due to the difficulty in accessing high-quality training data. For training this model, it was possible to use 3D bounding box annotations from the KITTI dataset [27]. The work presented in [72] used a more complex car model containing 12 vertices. This approach leverages keypoint-annotated datasets to lift the data from 2D to 3D, capturing intra-class shape variations by expressing each shape instance as a combination of a mean shape and basis vectors, thus allowing for a more flexible and accurate 3D shape estimation. In [123] Zhang et al. proposed the estimation of car models by training model using Fitness Evaluation Score (FES). This score measures how well a projected 3D vehicle model aligns with the image data by evaluating the gradient information of pixels within virtual rectangles formed around the visible projected line segments. The score considers the perpendicular gradient component of each pixel's magnitude and weights pixels closer to the projected line segment more heavily.

The second group of approaches aims to predict a dense 3D mesh of an object, providing a more detailed and comprehensive representation of its shape. Dense mesh prediction methods generally require more complex models and higher computational resources. These techniques often leverage advancements in deep learning architectures, such as variational autoencoders (VAEs) and generative adversarial networks (GANs), to generate detailed 3D meshes from input images. These dense mesh methods offer higher fidelity and can capture intricate details of the objects geometry, making them suitable for applications in 3D modeling and virtual reality [32].

Kundu et al. [51] encoded object shape using Principal Component Analysis (PCA) to a low-dimensional shape space. This representation uses a small set of parameters to describe 3D shapes, framing the shape estimation problem as predicting the appropriate set of low-dimensional shape parameters for a given object instance. In [38] to facilitate neural network training for shape reconstruction, Ke et al. reduced the shape representation dimension using PCA. They clustered models into four subsets using K-Means based on shape similarity. Firstly, they applied PCA for each subset to find n-dimensional shape representation. During inference, they predict the PCA coefficients for each cluster and then blend the final shape using weights acquired by the classification of input into one of four clusters. An approach from [53] uses the decomposition of an object shape into three components: mean shape, template offsets, and object offset. Then by using multi-head cross-attention, they predict object offsets to estimate the final object shape.

Graph Convolutional Networks (GCNs) have emerged as a powerful framework for learning on graph-structured data, combining elements of graph theory and deep learning to address a wide range of applications from social network analysis to bioinformatics and computer vision. The work of Kipf and Welling [43] introduced a simplified formulation of spectral graph convolutions, enabling efficient and scalable learning on graphs. Their approach, which approximates the spectral graph convolution using a first-order approximation of localised spectral filters, has become a cornerstone in the development of more complex GCN architectures. Subsequent work has extended this foundational model to tackle more dynamic and heterogeneous graphs. A significant contribution in this area is the work of Velikovi et al. [101], who introduced Graph

Attention Networks (GATs). These networks incorporate attention mechanisms into the GCN framework to improve model expressiveness and performance on node classification tasks. More recent advances have focused on improving the scalability and adaptability of GCNs to different types of graph data and structures, with the efforts of Wu et al. [115] presenting a comprehensive overview of different GCN variants and their applications in different domains. These contributions highlight the versatility and robustness of GCNs, making them a focal point of research in machine learning on data where additional information like local connections is available.

4.3 Estimation of 3D coordinates

This section will present an approach to estimating 3D keypoints of a vehicle from a single image. This module is responsible for estimating the 3D points in a canonical pose, which remains consistent regardless of the vehicle's observed pose. The canonical pose standardises the coordinate system such that its origin is located at the vehicle's geometric center, with the vehicle's front always oriented in a fixed direction.

The process begins with the feature maps generated from the image crops by the Image Backbone Network. In this experiment, HrNet [107] and ViT Pose [19] backbone networks were tested. These feature maps can be utilised also by the 2D Keypoint Head that predicts 2D keypoint coordinates on an image. Additionally, features extracted from the estimated 2D points are processed by the Keypoint Backbone Network. To form features from these 2D points, a Multilayer Perceptron (MLP) comprising seven layers is utilised. This MLP takes the normalised 2D coordinates, adjusted with respect to the bounding box, as its input. These derived feature maps, both from the image and the 2D points, are then concatenated to form the input for the Keypoint 3D Head (Fig. 4.1).

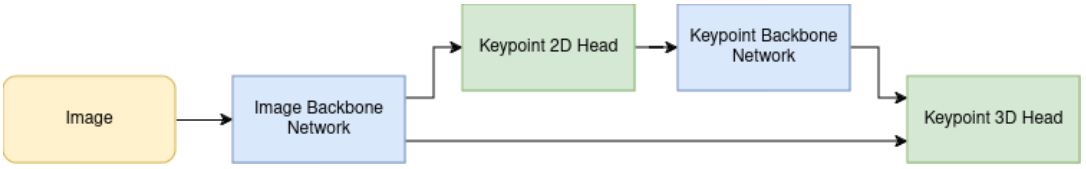

FIGURE 4.1: Pipeline for estimation of 3D keypoints.

Within the 3D Keypoint Head, two separate feature maps are estimated. The first feature map corresponds to the $X-Y$ plane, facilitating the estimation of the x and y coordinates of the vehicle in the canonical pose that corresponds to lateral and vertical axes. The second feature map corresponds to the $X-Z$ plane, which is used to estimate the z coordinate (longitudinal axis). This dual-map approach, which addresses two orthogonal planes independently, allows us to employ a similar processing pipeline to that used for 2D keypoint estimation. The used loss function is MSE as defined in Eq. (3.9). Moreover, the uncertainty of 3D keypoints can be estimated analogously as described in section 3.4 for keypoints 2D.

The design of the 3D point estimation module allows for seamless integration with the 2D point estimation network, as it reuses the feature maps extracted from the image. This integration not only streamlines the workflow but also enhances the overall efficiency and accuracy of the

pose estimation process by building upon the robust features derived during the 2D estimation stage. The experimental evaluation of this approach is presented in section 6.5.2.

4.4 Dataset preparation for supervised learning

For this experiment, the ApolloCar3D dataset, which was introduced in [93] was utilised. This dataset, sourced from the ApolloScape dataset [36], features a collection of high-resolution images (3384×2710 pixels), surpassing 140,000 semantically labeled images that depict complex driving scenarios. This dataset was curated with images chosen from labeled videos across four cities, emphasizing environments of relative complexity and ensuring a minimum interval of ten frames between selected images. To enrich diversity, the dataset was pruned manually to include images with a wide variety of car scales, shapes, orientations, and levels of occlusion among vehicles, resulting in a refined set of 5,277 images.

For 3D car models, the dataset demands high accuracy; models must align closely with manually labeled masks, with an offset boundary of less than 3 pixels on average. Given that existing 3D models from sources like ShapeNet were insufficiently precise, and considering the high cost of fitting each model in scenarios with significant occlusion, the project engaged professional model makers to construct bespoke 3D models. This dataset contains 34 CAD models that accurately represent the shape and scale of specific car types, including sedans, coupes, minivans, SUVs, and MPVs, covering a broad spectrum of commonly encountered vehicles.

Statistical data within the dataset reveal a significant variety of cars, often positioned at long distances or under heavy occlusion and showing diverse spatial distribution. The orientation data suggests that most cars are either moving towards or away from the data acquisition point. In terms of vehicle types, sedans appear most frequently. Importantly, many images in the dataset feature over ten labeled objects, underscoring the dataset's complexity and depth.

The dataset also incorporates an advanced semi-automatic keypoint annotation process that combines human annotators with machine assistance. ApolloCar3D defines 66 semantic keypoints per car but annotated are only those points that are visible from the camera's viewpoint. However, this keypoints set contains far more than any previous dataset and allows for more accurate and robust shape and pose registration. The location of keypoints annotated in this dataset is presented in Fig. 4.2. This approach ensures that ApolloCar3D not only supports detailed analyses of vehicle dynamics but also significantly enhances the development and testing of autonomous driving technologies. The visualization of keypoints 2D and cars' mesh is shown on Fig. 4.3

To address this issue, a procedure as illustrated in Fig. 4.5 was implemented. This method starts with a CAD model (Fig. 4.4A) that is transformed according to the provided translation and rotation parameters. By applying these transformations, it is ensured that the 3D model aligns accurately with the real-world pose of the vehicle captured in the image (Fig. 4.4B).

Next, the parameters of the rays are established, which include all the 3D points whose projections onto the image fall at the annotated keypoints of the vehicle (red dot on Fig. 4.4 B). For

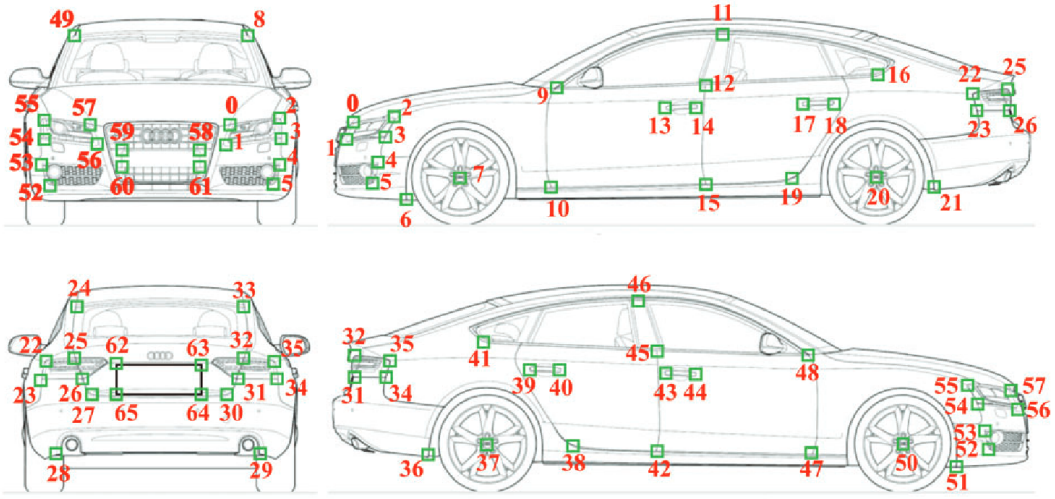

FIGURE 4.2: Location of 66 semantic keypoints annotated in ApolloCar3D Dataset. Source: [93]

FIGURE 4.3: Visualization of 2D keypoints (b) and 3D mesh (c) on an example frame (a) from ApolloCar3D Dataset. Source: [93]

each keypoint in the image, a ray is cast from the camera through the 2D keypoint location, extending into the 3D space of the transformed CAD model.

To determine where these rays intersect with the surfaces of the CAD model, the Möller-Trumbore algorithm [68] was utilised. This algorithm is an efficient method for detecting ray-triangle intersections, which is particularly useful given the triangular mesh structure of the CAD models. By applying the Möller-Trumbore algorithm, intersections between the rays and each face of the CAD model are systematically checked. When an intersection is found, the precise coordinates of the intersection point on the model's surface are calculated (Fig. 4.4C).

In scenarios where multiple intersection points are identified, the point closest to the camera is selected (green dot on Fig. 4.4C, yellow dot will be rejected because it is further from the camera). This strategy aligns with the dataset's methodology of annotating only the non-occluded points and allows to inclusion of only correct matches between 2D and 3D points.

The final stage of this procedure involves transforming the intersection points back to the car's canonical pose. This is achieved by applying the inverse of the ground truth rotation and translation transformations to the identified intersection point. This transformation yields the coordinates of the keypoints with respect to the standardised orientation and position of the car model, facilitating consistent and uniform data representation.

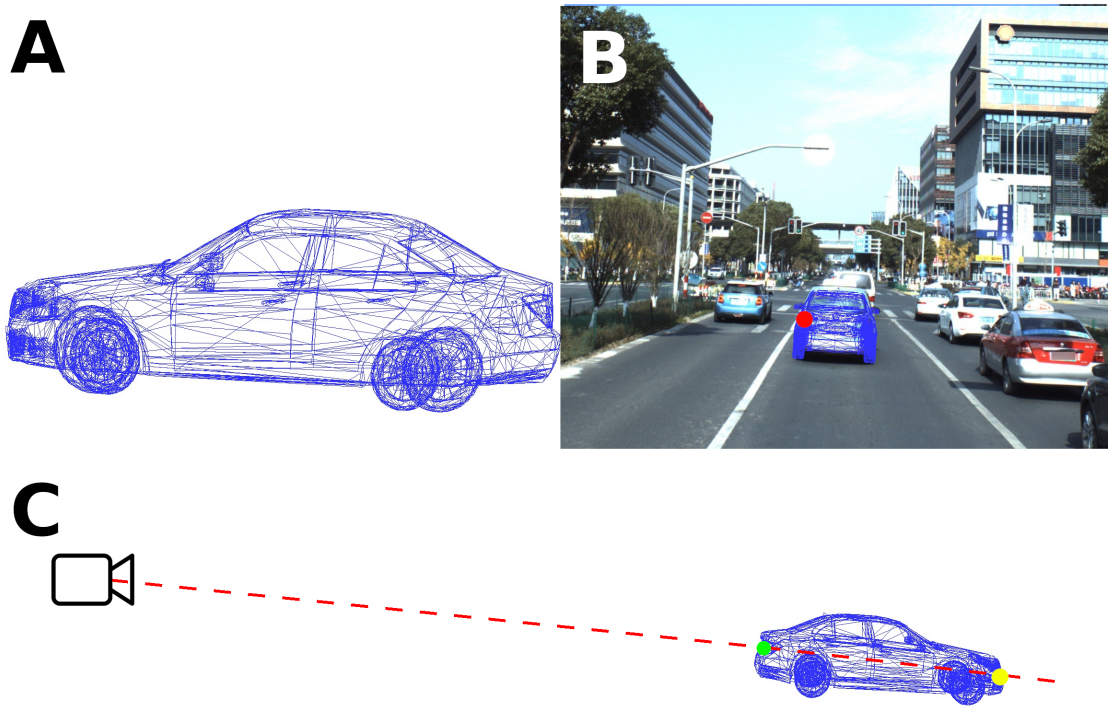

FIGURE 4.4: Visualization of keypoints coordinates estimation using rays and mesh intersection.

Following the initial mapping and transformation, a fine-tuning process is conducted to enhance the precision of the derived coordinates. For this fine-tuning step, the Nelder-Mead optimization algorithm, which is applied to all instances of a particular car type within the training set was utilised. This algorithm iteratively adjusts the 3D coordinates to minimise the translation

error. The translation error is quantified as the square of the distance between the ground truth translation and the translation estimated by the Efficient Perspective-n-Point (EPnP) [54] method. The EPnP method takes into account the provided camera parameters and the 2D keypoint coordinates in the image, ensuring that the optimization is guided by accurate geometric and camera model considerations.

By employing the Nelder-Mead optimization, the initial estimates were refined, achieving a more precise alignment between the 2D and 3D keypoints. This fine-tuning process enhances the accuracy of the keypoint localization. The resulting optimised coordinates create a high-quality dataset that effectively supports the training and validation keypoint 3D estimation models.

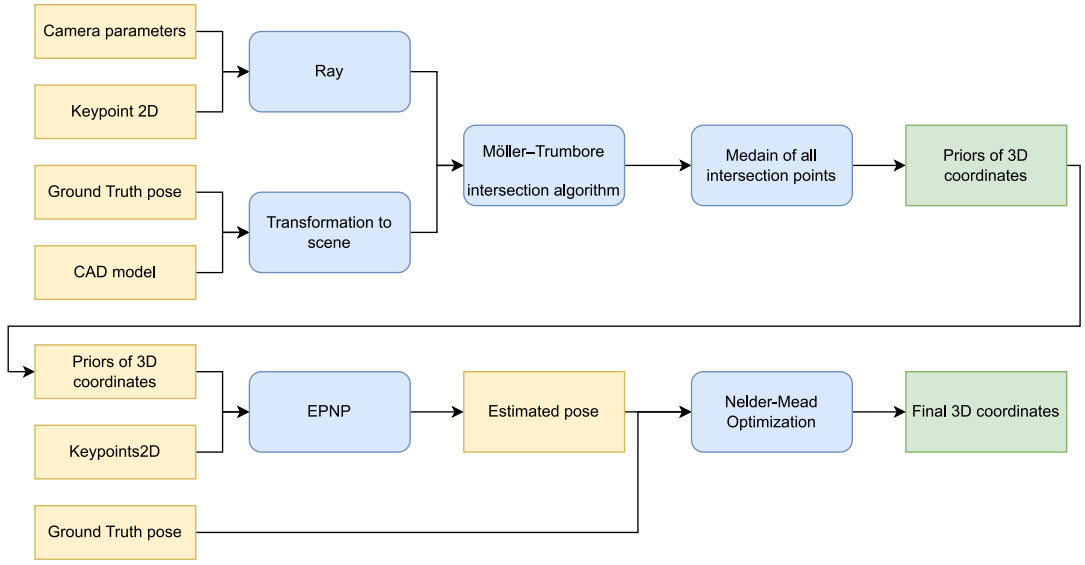

FIGURE 4.5: Pipeline for dataset pre-processing in order to obtain labelled 3D points for supervised learning.

4.5 Reprojection loss

For the training of 2D and 3D keypoint heads, the reprojection loss function was applied. This function ensures the geometric consistency of the estimated coordinates. It operates by comparing the projected 3D points, transformed using ground truth data, with the predicted 2D points. This comparison ensures that the network's estimations of 2D and 3D points are aligned and consistent with each other, as well as with real-world geometries. By enforcing this consistency, the reprojection loss function helps the network maintain accurate spatial relationships between the estimated points, which is needed for reliable 3D shape estimation from 2D images. This loss is defined as:

$$\mathcal{L}_{\text{reprojection}} = \sum_{i=1}^{n} \left(\left\| \pi \left(\mathbf{T}, \hat{\mathbf{p}}_i^{3d}, \mathbf{K} \right) - \hat{\mathbf{p}}_i^{2d} \right\|_2 \right)^2, \tag{4.1}$$

where π is the projection function, $\mathbf{T}$ is a ground truth pose, $\hat{\mathbf{p}}_i^{3d}$ are the estimated 3D coordinates of the i-th characteristic point, $\mathbf{K}$ is the camera intrinsics matrix, and $\hat{\mathbf{p}}_i^{2d}$ are the estimated 2D coordinates of the i-th keypoint on image. The application of this reprojection loss function improves the accuracy of the point estimation heads. By enforcing geometric consistency between

the predicted 2D points and their corresponding projected 3D points, the network is able to refine its estimations.

4.6 Estimation of 3D model

This section presents an approach to the estimation of dense object mesh from a single image. A network that estimates a dense mesh of vehicles offers several advantages over networks that directly estimate 3D keypoints. By reconstructing the full vehicle mesh, this approach allows for a better understanding of the vehicle's geometry, enabling more precise extraction of semantic keypoints. This precision arises because the dense mesh encompasses all relevant geometric details, ensuring that keypoints derived from this model better reflect the vehicle's structure. Consequently, the derived keypoints are inherently aligned with the true geometric properties of the vehicle, leading to more reliable inputs for algorithms solving Perspective-n-Point problem and, subsequently, more accurate pose estimations. Additionally, the dense mesh approach supports robust performance against partial occlusions and varying lighting conditions, which can often challenge methods that estimate keypoints directly. This adaptability enhances the network's utility across different operational scenarios, thus providing a better tool for vehicle pose estimation tasks. A detailed 3D mesh of surrounding vehicles allows autonomous systems to perform precise spatial analyses, needed, e.g. collision detection. By understanding the exact shape and size of nearby vehicles, an autonomous driving system can predict potential collisions more accurately and make more informed decisions about maneuvers in order to avoid accidents.

This pipeline takes as input the image crop containing the car, with the expected output being a mesh that represents the car's shape. The construction of this pipeline involves two steps. The first step is the training of a Variational Autoencoder capable of encoding a car's shape to a latent vector, the second step is training a network for estimation of mesh from image input Fig. 4.6.

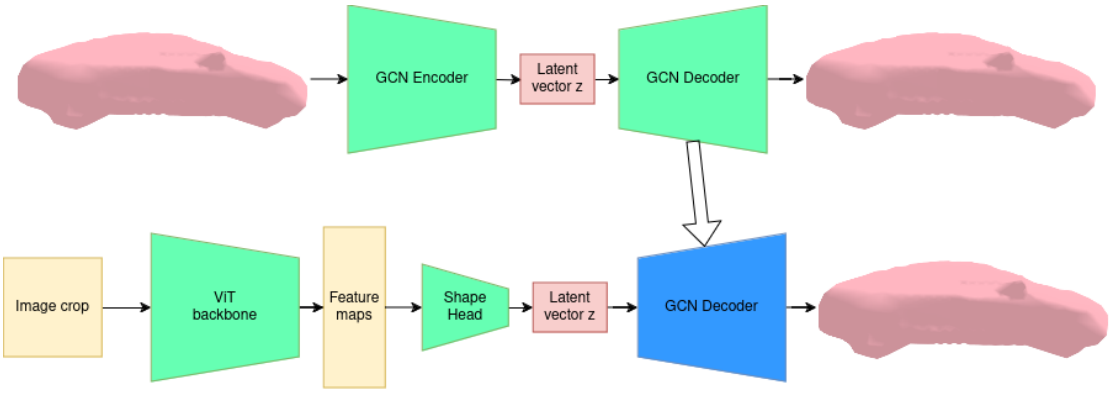

FIGURE 4.6: Architecture of the pipeline for the car shape estimation. Green blocks show trainable parts of the network, and the blue blocks mean that the weights are frozen during the training phase.

The Variational Autoencoder architecture employs Graph Convolutional Networks designed for encoding and decoding the 3D shape of cars. The input for the network is a mesh containing

vertices and the edges - connections between the closest vertices and the desired output is the same mesh as provided in the input.

The dataset used for training the VAE was prepared by combining car meshes from the Apollo Car3D dataset [93] and meshes acquired from the VUMO company [103]. An important aspect of this process is ensuring the uniformity of the mesh topology. Uniformity, in this context, refers to the consistency in the number of mesh vertices and the local connections between them (edges). To achieve this, the raw meshes were processed to meet these uniformity requirements.

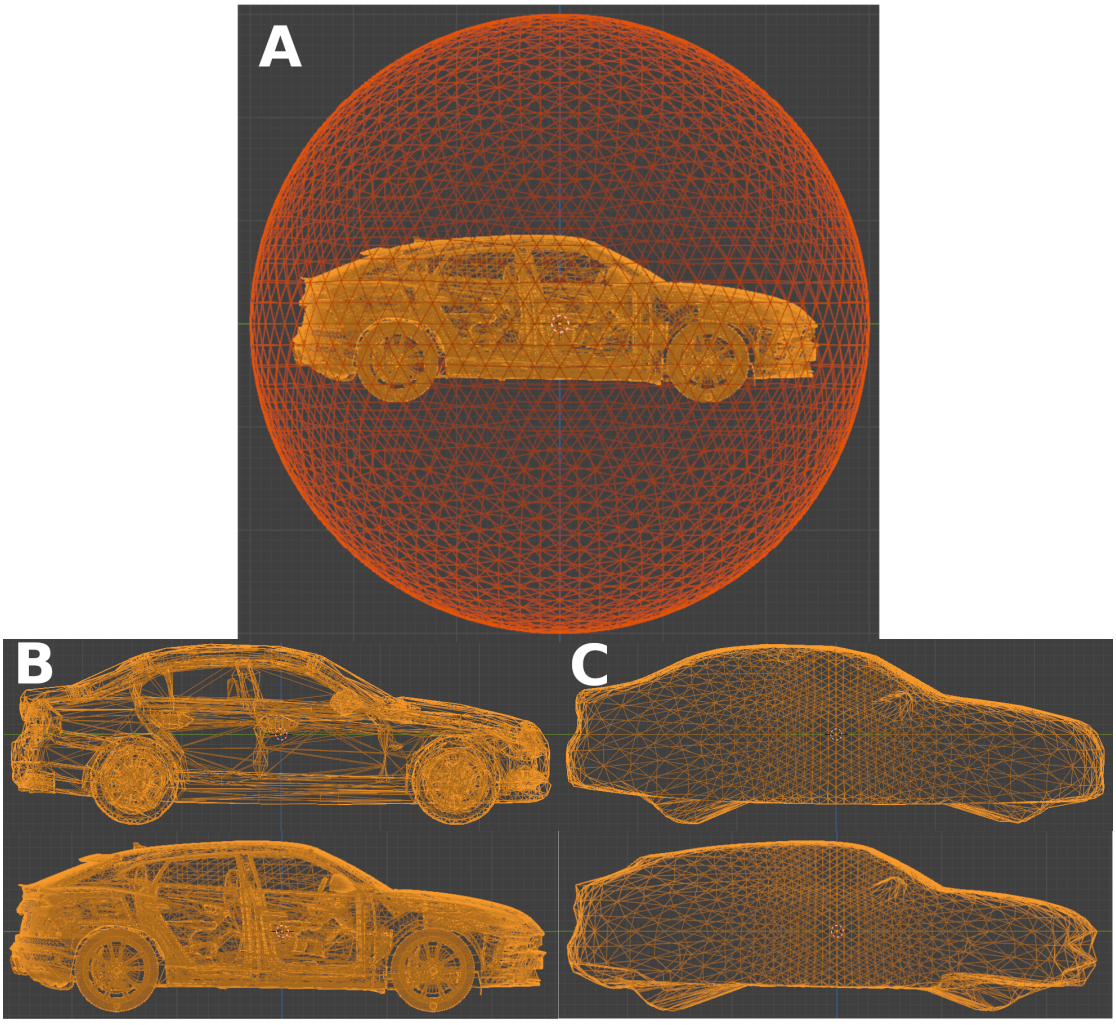

FIGURE 4.7: An icosphere and input car mesh before warping procedure (A). Examples of raw car meshes from datasets (B) and processed meshes with uniform topology (C)

The creation of processed mesh begins with an icosphere containing 2562 vertices (Fig. 4.7A). This icosphere is then adjusted to wrap around the car's model, ensuring it conforms to the shape of the car. Examples of input meshes and processed, uniform meshes are shown in Fig. 4.7B and C. The final dataset, used for training, comprises 136 distinct car meshes, each with a uniform topology. Examples are presented in Fig. 4.8.

The encoder part of the VAE utilises a sequence of 16 GCNConv layers, designed to process the graph representation of a car's shape, which comprises vertices and their edge connections. The output channels for these layers are set to the following values: [12, 48, 96, 192, 384,

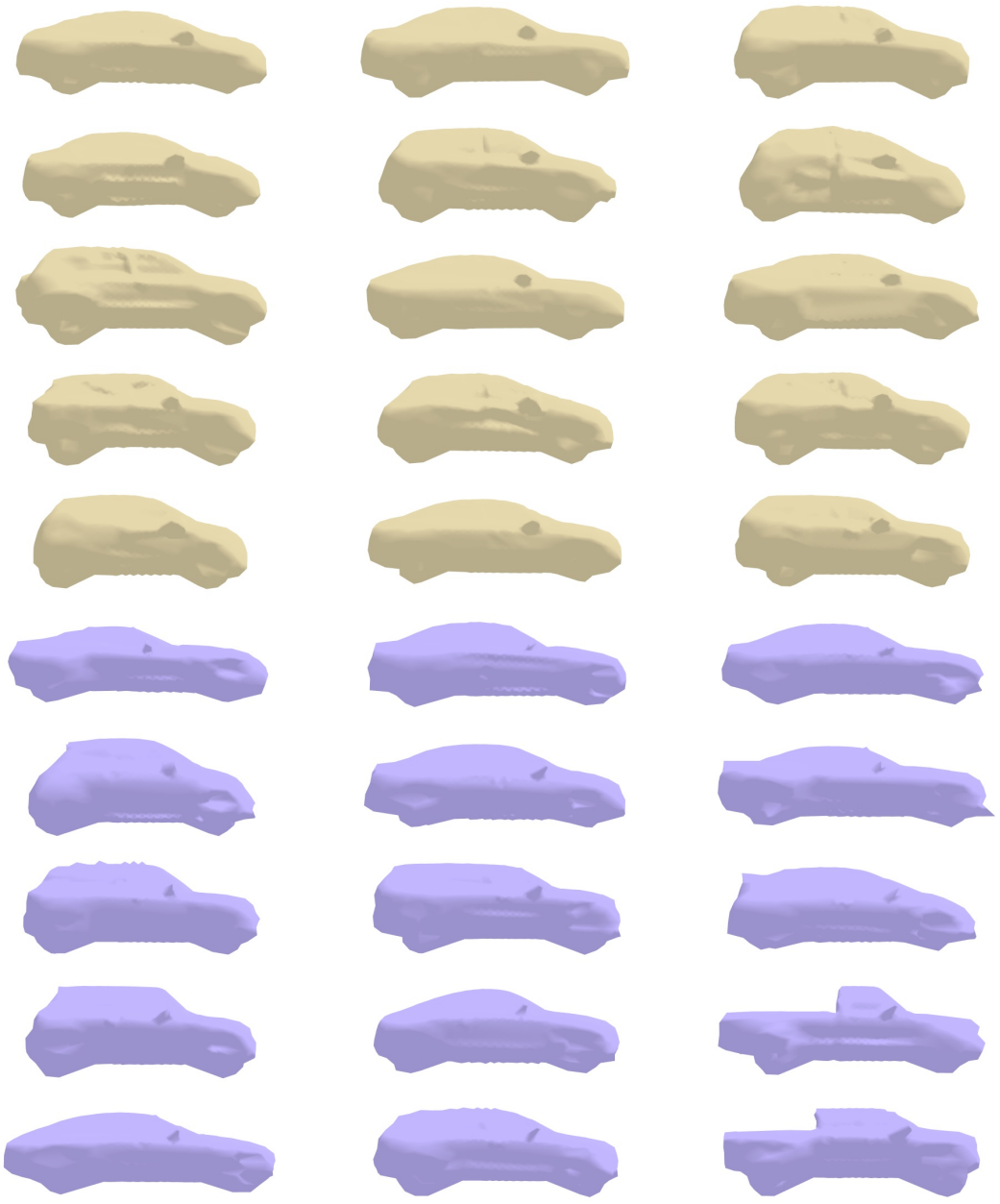

FIGURE 4.8: Examples of car meshes from train dataset. Yellow examples are from the ApolloCar3D dataset and blue cars are obtained from VUMO.

768, 1536, 1536, 3072, 768, 192, 96, 48, 48, 12, 1], allowing for a detailed feature extraction process that gradually abstracts the input into a more informative, high-level representation. This structured layering ensures the model captures a comprehensive spectrum of features from simple to complex, necessary for precise shape encoding.

After GCN processing, the model transitions to a series of linear layers to compress the mesh representation into a latent vector. Starting with the input size of 2562, corresponding to the number of mesh vertices, the process involves several stages of dimension reduction through linear layers and LeakyReLU activations. The sizes progressively reduce through a calculated sequence: from 2562 through 640, 320, 160, to a latent size of 32.

The decoder of this VAE is engineered to reconstruct the input mesh from the encoded latent vector, effectively reversing the encoding process to regenerate the 3D shape of a car from its abstracted form. This process initiates with the latent vector traversing a sequence of expansive linear layers that incrementally increase the data's dimensionality. The data is progressively doubled in size through a sequence of linear transformations paired with LeakyReLU activations, culminating in a dimension that matches the total number of output channels, equivalent to the number of vertices in the input mesh. The number of output features of each linear layer is [64, 128, 256, 512, 1024, 2562].

Following the linear transformations, the data is reshaped to fit the requirements of the Graph Convolutional Network layers. These layers are essential for mapping the high-dimensional linear output back into the structured format of a mesh. The reshaped data undergoes processing through multiple GCN layers. The GCN layers start from a single channel and progressively build up complexity through a series of graph convolution operations paired with LeakyReLU activations. The exact values of output channels are: [3, 12, 24, 96, 192, 384, 768, 1536, 1536, 3072, 768, 192, 96, 48, 12, 3]. A VAE network architecture is shown in Fig. 4.9.

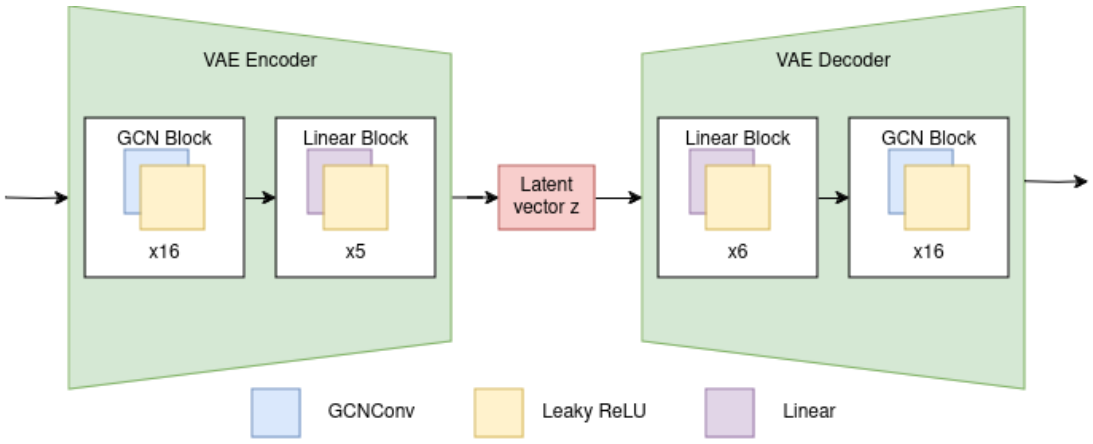

FIGURE 4.9: Architecture of the car's mesh Variational Autoencoder.

During training, three loss functions are employed to ensure accurate mesh estimation. The first, loss $\mathcal{L}_{\text{MPVPE}}$, measures the mean vertex estimation error, providing a direct assessment of how closely the estimated vertices match the true vertices. The second loss $\mathcal{L}_{\text{Dim}}$, measures the discrepancy in the external dimensions of the car, ensuring that the overall size and proportions of the estimated mesh match the actual car. The third loss $\mathcal{L}_{\text{KLD}}$, measures the Kullback-Leibler Divergence (KLD), which regularises the latent space by ensuring that the distribution of the latent vectors remains close to a normal distribution. The loss function is formulated in Equations: (4.3), (4.4), (4.5) and (4.6).

$$\mathcal{L}_{\text{total}} = \lambda_{\text{MPVPE}}\mathcal{L}_{\text{MPVPE}} + \lambda_{\text{Dim}}\mathcal{L}_{\text{Dim}} + \lambda_{\text{KLD}}\mathcal{L}_{\text{KLD}}, \tag{4.2}$$

where $\mathcal{L}_{\text{total}}$ is the total loss function, λ_{MPVPE} is the scaling weight for the Mean Per Vertex Position Error (MPVPE) loss, λ_{Dim} is the scaling weight for the dimension discrepancy loss, λ_{KLD} is the scaling weight for the KLD loss.

$$\mathcal{L}_{\text{MPVPE}} = \frac{1}{N}\sum_{i=1}^{N} \left\|\hat{\mathbf{v}}_i - \mathbf{v}_i\right\|_2 , \tag{4.3}$$

where N is the number of vertices, $\hat{\mathbf{v}}_i$ represents the estimated vertex position, and $\mathbf{v}_i$ represents the ground truth vertex position.

$$\mathcal{L}_{\text{Dim}} = \left|\hat{l} - l\right| + |\hat{w} - w| + \left|\hat{h} - h\right| , \tag{4.4}$$

where $\hat{l}$ is the estimated length of the car, $\hat{w}$ is the estimated width of the car, $\hat{h}$ is the estimated height of the car, l is the ground truth length of the car w is the ground truth width of the car and h is the ground truth height of the car.

$$\mathcal{L}_{\text{KLD}} = -\frac{1}{2}\left(1 + \log(\sigma^2) - \mu^2 - \sigma^2\right) , \tag{4.5}$$

where μ is the mean of the latent variable and σ is the standard deviation of the latent variable.

The architecture of the network for car shape estimation is depicted in the lower row of Fig. 4.6. The pipeline begins with an image crop containing the car as input. This image is processed by the Vision Transformer backbone network, which extracts feature maps. These feature maps are then processed by the Shape Head. The primary task of the Shape Head is to estimate the latent vector z that best describes the car under consideration. Once this latent vector z is estimated, the decoder part of the VAE uses it to generate the full car's mesh.

The Vision Transformer used for feature extraction, was implemented in mmpose framework [67]. The used variant has the following configuration. This network processes images of size 192x256 pixels by breaking them down into patches of 16x16 pixels each. An embedding dimension equal to 1024, which means that each patch representation processed by the transformer is a vector of 1024 values. The architecture comprises 24 layers. Each layer employs 16 attention heads. The transformer utilises feedforward networks within each transformer block, with max channels set to 4096. Additionally, the network incorporates a drop path rate of 0.3 to combat overfitting during training by randomly dropping out certain paths in the attention mechanisms. The initial training weights have been pretrained on the ImageNet dataset. The Shape Head is built by 4 convolutional layers with a number of output filters equal to: [512, 256, 128, 1]. At the end, there is a single linear layer that outputs 32 values that build latent vector z. The final mesh is obtained from the output of the VAE Decoder block.

Once the dense mesh of the car has been estimated, it becomes possible to determine the coordinates of all characteristic points on the vehicle. This task was accomplished by utilising a straightforward three-layer MLP, which was trained in conjunction with the rest of the network.

During training, the weights of the VAE Decoder block were frozen. The loss function was similar to Eq. (4.2) and described by Eq. (4.6). The only difference was an additional component $\mathcal{L}_{\text{Kpts3D}}$. That minimised the error between the predicted and ground truth coordinates of the 3D keypoints (Eq. (4.7)).

$$\mathcal{L}_{\text{total}} = \lambda_{\text{MPVPE}}\mathcal{L}_{\text{MPVPE}} + \lambda_{\text{Dim}}\mathcal{L}_{\text{Dim}} + \lambda_{\text{KLD}}\mathcal{L}_{\text{KLD}} + \lambda_{\text{Kpts3D}}\mathcal{L}_{\text{Kpts3D}}, \tag{4.6}$$

where $\mathcal{L}_{\text{total}}$ is the total loss function, λ_{MPVPE} is the scaling weight for the MPVPE loss, λ_{Dim} is the scaling weight for the dimension discrepancy loss, λ_{KLD} is the scaling weight for the KLD loss and λ_{Kpts3D} is the scaling weight for the Keypoints3D loss.

$$\mathcal{L}_{\text{Kpts3D}} = \frac{1}{n}\sum_{i=1}^{n} \left\| \hat{\mathbf{p}}_i^{3d} - \mathbf{p}_i^{3d} \right\|_2, \tag{4.7}$$

where n is the number of characteristic points, $\hat{\mathbf{p}}_i^{3d}$ represents the estimated keypoint position, and $\mathbf{p}_i^{3d}$ represents the ground truth keypoint position. The experimental results of this network are discussed in section 6.6.

Chapter 5

Camera pose estimation

5.1 Introduction

Estimation of the pose of surrounding objects is an important component for understanding the vehicle environment and ensuring safety in autonomous driving. This chapter explores the concept of pose estimation, which involves determining the position and orientation of objects relative to the vehicle using a monocular camera. Accurate pose estimation is required for effective navigation, influencing the autonomous vehicles decision-making capabilities, and impacting tasks such as obstacle avoidance, path planning, and interaction with other road users. This task involves the detection and 3D positioning of various entities like other vehicles, pedestrians, and road infrastructure objects. The precision of this process directly affects the vehicle's ability to interpret its surroundings and respond appropriately. For instance, having precise positions of surrounding cars allows for the estimation of their velocity by tracking how their positions change over time. This approach requires robust algorithms to accurately monitor the positions and movements of other vehicles. By ensuring these algorithms are precise and reliable, autonomous vehicles can better predict the behavior of surrounding objects, enhancing overall safety and efficiency in navigation.

Despite its importance, camera-based pose estimation in autonomous driving faces several challenges. Variations in environmental conditions, such as lighting and weather, pose significant difficulties. Additionally, the presence of dynamic and partially occluded objects in driving scenarios complicates the estimation process. A critical requirement for autonomous vehicles is the need for real-time processing in pose estimation. This necessitates the development of algorithms that are not only accurate but also computationally efficient.

There are works that use LIDARs for the pose estimation task. This technology is often used in autonomous vehicles. However, there are notable drawbacks to LiDAR-based pose estimation. Firstly, LiDAR systems tend to be quite expensive, which can increase the cost of manufacturing autonomous vehicles. Additionally, while LiDAR is excellent for distance measurement, it lacks the color and texture detail provided by optical cameras, needed for recognising and classifying objects and provides relatively low resolution. Another limitation is performance in adverse

weather conditions. LiDAR sensors can struggle in heavy rain, fog, or snow, as these conditions can scatter the laser beams and degrade the accuracy of the sensors readings.

There are two primary approaches to visual pose estimation:

- The stereovision method, which mimics human depth perception by using two cameras at different viewpoints, has traditionally been used in pose estimation. This method calculates depth information by comparing the disparity between the images from the two cameras. The advantage of stereovision lies in its accuracy in depth determination, which is important for understanding the 3D structure of the environment. However, the implementation of stereovision in vehicles comes with challenges. It involves the calibration and synchronization of two cameras, which adds to the system's complexity and cost.
- The monocular approach, which uses a single camera for pose estimation. This method has gained popularity due to advancements in computational techniques, particularly in the fields of machine learning and artificial intelligence. The monocular approach is simpler and more cost-effective, as it requires only one camera and minimal calibration. However, the challenge with this approach is the accurate estimation of depth from a single image, a task that is inherently more complex than with stereovision. Recent developments in deep learning have, however, improved the efficacy of monocular systems, making them a viable option for pose estimation in autonomous vehicles [6].

The main approach to estimating pose from a single camera relies on methods that solves the Perspectve-n-Points problem. Iterative methods for solving this problem, are highly adaptable solutions. They leverage iterative optimization techniques to refine an initial pose estimate through successive minimization of a chosen objective function, usually the reprojection error. Their customizable nature allows for different optimization algorithms to be employed based on the specific requirements of the problem. Additionally, they offer the flexibility to incorporate bounds and constraints on the optimization process to ensure physically plausible solutions. Moreover, users can apply various loss functions tailored to the application's needs.

The selection of accurate and reliable points is important for the effectiveness of PnP algorithms. Points that are precisely matched and broadly distributed across the target object enable more stable and well-defined estimations, reducing uncertainties in camera pose calculations. On the other hand, using points that are poorly distributed or inaccurately matched can lead to significant errors, resulting in incorrect pose estimates. Additionally, outliers or points obscured by occlusions can skew the results if they are not carefully excluded [77].

The need for real-time processing in autonomous vehicles significantly impacts the development of pose estimation technologies. These vehicles must analyse a large amount of sensory data, including information from pose estimation systems, almost instantaneously for safe and effective navigation. Stereovision systems process two synchronised video streams, which adds complexity. Monocular systems, though simpler in hardware, rely on complex software algorithms, often using artificial intelligence to ensure precision. These algorithms are designed to optimise the trade-off between accuracy and processing speed, as delays can result in the use of outdated information, affecting the vehicles ability to adapt to changing road conditions. The challenge is to develop

pose estimation methods that are both accurate and fast enough to support the vehicle's rapid decision-making needs.

Understanding the uncertainty in pose estimation is important for autonomous vehicles, which rely on their perception systems to make driving decisions. The effectiveness of these systems in accurately perceiving the vehicles environment impacts their navigation safety. Pose uncertainty estimation provides a probabilistic measure of confidence in the vehicle's pose estimates. This information helps in navigating complex environments, where accurate knowledge of the vehicle's position relative to other objects such as pedestrians, vehicles, and road infrastructure is necessary for safe maneuvering and collision avoidance.

Estimating uncertainty, especially when using Convolutional Neural Networks, presents challenges. CNNs are widely used for image processing tasks and are fundamental to many autonomous vehicle perception systems. However, these networks usually do not offer insights into the confidence level of their predictions. This limitation is important in autonomous driving, where decisions based on uncertain predictions can affect safety. Therefore, developing CNN architectures that predict pose, and quantify the uncertainty of these predictions requires new approaches in network design and training.

The practical applications of accurately estimated pose uncertainty in autonomous vehicles are extensive. It improves the safety of these vehicles by offering a measure of the reliability of pose estimations, which supports better risk assessment and decision-making in uncertain environments. This is important in situations marked by ambiguous or incomplete information, like in adverse weather or on unstructured roads. Furthermore, the quantification of uncertainty is important in sensor fusion algorithms, where data from various sensors such as cameras, LiDAR, and Global Positioning System (GPS) are combined to understand the vehicle's surroundings. Accurate uncertainty estimation enhances the calibration of these algorithms, ensuring that the most reliable data influences decision-making processes.

Moreover, understanding pose uncertainty can aid in the development of adaptive control systems for autonomous vehicles. These systems can dynamically adjust the vehicles behavior based on the level of confidence in the pose estimation, ensuring optimal performance under varying conditions. For instance, in situations where the uncertainty is high, the vehicle might adopt a more cautious driving strategy.

The estimation of pose uncertainty also has broader implications in the development of autonomous driving technologies. It contributes to the robustness of autonomous systems, making them more resilient to unexpected changes in the environment or sensor malfunctions [108].

In this chapter, the focus is on the problem of pose estimation using a monocular camera, addressing two specific scenarios within the context of autonomous driving. The first scenario involves estimating the pose of a city bus relative to a charging station. This application is designed to assist drivers or autonomous systems in accurately docking buses to charging stations, a task that requires precise alignment to ensure successful connection and charging. The second scenario deals with estimating the pose of surrounding vehicles in an urban environment. This task is required for navigating complex traffic scenarios where understanding the orientation and position of other vehicles is necessary for safe and efficient driving decisions. Both scenarios

pose unique challenges due to the reliance on a monocular camera, which provides limited depth information. Additionally, the field of autonomous driving demands not only accurate pose estimation but also a quantification of the uncertainty associated with these estimates. This requirement highlights the need for advanced methodologies capable of delivering both precise pose estimations and corresponding confidence assessments, ensuring that autonomous systems can make informed decisions despite uncertainty.

This chapter presents the following contributions to the field of pose estimation in the context of autonomous driving:

- A full processing pipeline for estimation of the pose of surrounding cars from a single monocular image
- A pose refinement procedure for correction of pose estimation error caused by outlier characteristic point estimations
- An application of unscented transform for propagation of keypoints uncertainty for the uncertainty of estimated pose

5.2 Related work

A common approach to pose estimation is to use algorithms that solve the Perspective-n-Point problem, calculating the orientation and position of a camera in 3D space by matching 2D image points with known 3D coordinates. Over the years, several algorithms have been developed to address this problem, each offering unique advantages and optimizations for different scenarios.

A significant contribution to the field of pose estimation is the EPnP algorithm developed by Lepetit et al. [54]. The EPnP is advantageous due to its computational efficiency and the ability to provide accurate pose estimates with a minimal set of points, as few as four. This efficiency makes it particularly useful in real-time systems.

The Perspective-Three-Point (P3P) problem focuses on determining the position and orientation of a camera given only three known reference points in 3D space and their corresponding 2D projections in the image. The paper by Xiao et al. [25] offers a solution to this problem through both algebraic and geometric approaches. Using Wu-Ritt's zero decomposition algorithm, the authors present a complete triangular decomposition of the P3P equation system, providing the first comprehensive analytical solution. The geometric approach complements this by offering purely geometric criteria to determine the number of real-world solutions, adding valuable depth to the problem's understanding. In [44] Kneip introduced a novel, closed-form solution to this problem, directly computing the camera's position and orientation in one step without intermediate point derivation. By utilising an intermediate frame of reference and two-parameter representation, the proposed approach significantly improves computational efficiency and numerical stability, offering solutions up to 15 times faster than existing state-of-the-art methods at that time.

Different types of methods for solving the PnP problem are iterative algorithms [60], which are known for their robustness and accuracy in handling real-world data. Those methods iteratively

refine the camera pose estimate to minimise the reprojection error, which is the distance between the projected 3D points and the corresponding 2D image points. Multiple optimisation methods can be applied within those algorithms.

The Gauss-Newton [26] is a widely used algorithm for nonlinear least squares problems. The Levenberg-Marquardt [70] algorithm effectively combines the gradient descent method, which adjusts model coefficients in the direction that most reduce error, and the Gauss-Newton method, which approximates the least squares function as locally quadratic, adapting its approach based on how close the current solution is to the optimal. The Broyden-Fletcher-Goldfarb-Shanno (BFGS) [23] computes the descent direction by enhancing the gradient with curvature data, incrementally refining an estimate of the Hessian matrix of the loss function. This estimate is derived solely from evaluations of the gradient (or its approximations) through a generalised secant method. Lastly, the Nelder-Mead [73] algorithm, used in the previous chapter, provides a robust alternative for optimization problems where derivatives are not available.

In recent years, deep learning-based methods have revolutionised monocular pose estimation by learning rich feature representations directly from data. One prominent approach is to use convolutional neural networks to extract features from a single image and then regress the 3D pose of objects. For instance, Mousavian et al. [71] introduced a method for 3D bounding box estimation from a single image. 3D information is estimated based on the assumption that the perspective projection of a 3D bounding box should fit tightly within its 2D detection window.

Another contribution in the specific context of autonomous driving is the work by Chen et al. [15], which proposed a method that utilises energy minimization to position object candidates in 3D, ensuring they align with the ground plane. Each candidate box, when projected onto the image plane, is scored based on multiple factors including semantic segmentation, contextual information, size and location priors, and typical object shape.

The 6D-VNet extends Mask R-CNN by incorporating customised heads for predicting the vehicle's finer class, rotation, and translation. 6D-VNet is trained end-to-end, which significantly simplifies the training process and enhances performance. The inclusion of translational regression in the joint losses addresses the significant variations in object translation distances along the longitudinal axis common in autonomous driving scenarios. Furthermore, 6D-VNet incorporates mutual information between traffic participants via a modified non-local block, which considers spatial neighboring information [113].

Filtering outliers has its application in many computer vision applications, including (PnP) algorithms, where even a few erroneous data points can significantly distort results. Outliers often emerge due to errors in data processing or occlusions. The filtering process typically involves leveraging statistical methods or robust optimization techniques to detect and discard these outliers before solving the main problem. For instance, RANSAC (Random Sample Consensus) [22] is a popular method that iteratively tests subsets of the data to identify a consensus set that represents the majority of points. By focusing on inliers, the algorithm ensures that subsequent computations produce accurate estimations of the camera pose or scene geometry. This paper [7] introduced Graph-Cut RANSAC (G-C RANSAC). Its local optimization step is globally optimal for the best model parameters. The authors also proposed a criterion for applying this step,

which significantly improves processing time without compromising accuracy. GC-RANSAC can be combined with features like Progressive Sample Consensus (PROSAC) sampling.

5.3 Pose estimation of objects with known shape and dimensions

One of the simpler methods for estimating the position of known objects is based on the size of the bounding box in an image. This method relies on the proportions between the object's projection on the image and its actual dimensions. It requires knowledge of the camera's intrinsic parameters and can only be applied when the dimensions of the object are known and the perspective from which the object is viewed is limited. An example scenario is docking with an electric city bus to the charging station.

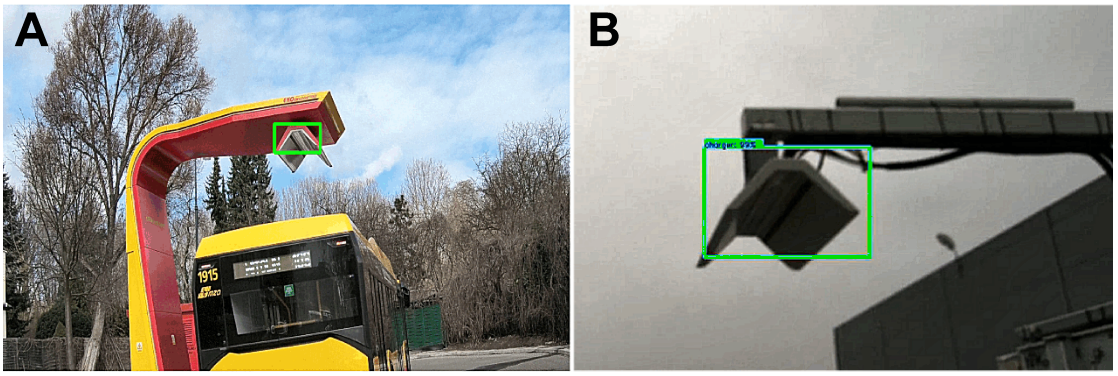

FIGURE 5.1: Examples of Faster R-CNN detection: correct bounding box (A), and oversized bounding box (B)

Figure 5.2 shows the geometric construction that is used to compute the distance having only the dimensions of the bounding box. The coordinates x, y and θ define the position and orientation of the charger roof with respect to the camera coordinate system.

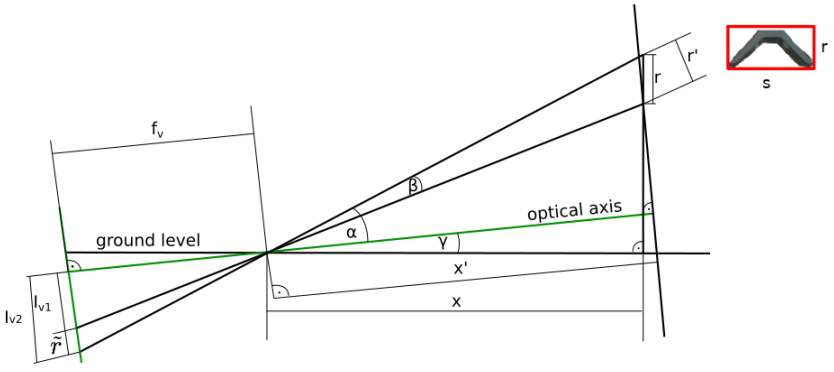

FIGURE 5.2: Visualization of distance calculation in x axis

For the calculation of the distance from the camera to the charger, it is necessary to have information about the dimensions of the bounding box surrounding the charger, the intrinsic parameters of the camera as determined by calibration, and the physical measurements of the charger itself. The positions of pixels within an image are mapped out using the (u, v) coordinate system.

By specifying (f_u, f_v) to represent the focal lengths of the camera along the u and v axes respectively, (c_u, c_v) to indicate the central point of the camera's image, and r and s to denote the actual height and width of the charger, the foundation for our computations was set. Furthermore, $\tilde{r}$ and $\tilde{s}$ refer to the height and width of the charger's image projection measured in pixels, while I_{v1} and I_{v2} identify the lower and upper edges of the charger's projection in relation to the image's central axis. I_u marks the position of the charger's center on the image along the u axis. With these parameters established, the angles α and β are determined, enabling the measurement of the distance by leveraging the geometric relationships and projections captured by the camera:

$$\alpha = \arctan\left(\frac{I_{v2}}{f_v}\right), \quad \beta = \alpha - \arctan\left(\frac{I_{v1}}{f_v}\right). \tag{5.1}$$

Then, having a known camera pitch angle (γ) the translation can be estimated from the equations:

$$x = \frac{f_v \cdot r}{\tilde{r}} \cdot \frac{\cos(\alpha + \gamma)}{\cos\alpha} \cdot \frac{\cos(\alpha - \beta + \gamma)}{\cos(\alpha - \beta)}, \quad y = \frac{x \cdot (I_u - c_u)}{f_u}. \tag{5.2}$$

The process of identifying the pitch angle involves aligning it with the direction of the gravity vector, a task accomplished by utilising the Inertial Measurement Unit (IMU) that is calibrated with the camera.

Bounding boxes, while commonly used in object detection, often fail to provide useful additional information for improving estimation accuracy. They typically encompass more area than the actual projection of the object on the image, as illustrated in (Fig. 5.1B), leading to inefficiencies in precise pose estimation. To address these shortcomings, the capabilities of the Mask R-CNN network have been utilised. This approach leverages detailed silhouette information of the charger (Fig. 5.3A), which aids in determining the θ angle. However, this method is sensitive to inaccuracies; even minor variations in the detected shape, such as those caused by changes in lighting, can result in significant deviations in angle estimation.

The output from the Mask R-CNN is a grayscale patch that matches the size of the detection bounding box, where each pixel's intensity indicates the likelihood of that pixel being part of the charger. The application of a threshold to this grayscale patch facilitates a clearer delineation of the charger's silhouette.

Another approach to this problem involves using an algorithm that addresses the Perspective-n-Point problem. The pose estimation in this method relies on detecting keypoints using a deep learning framework and incorporating the knowledge of a 3D model of the object. Early tests showed that the best results were achieved using an solvePnP iterative algorithm [60] designed to minimise the reprojection error, which is calculated as the sum of squared distances between

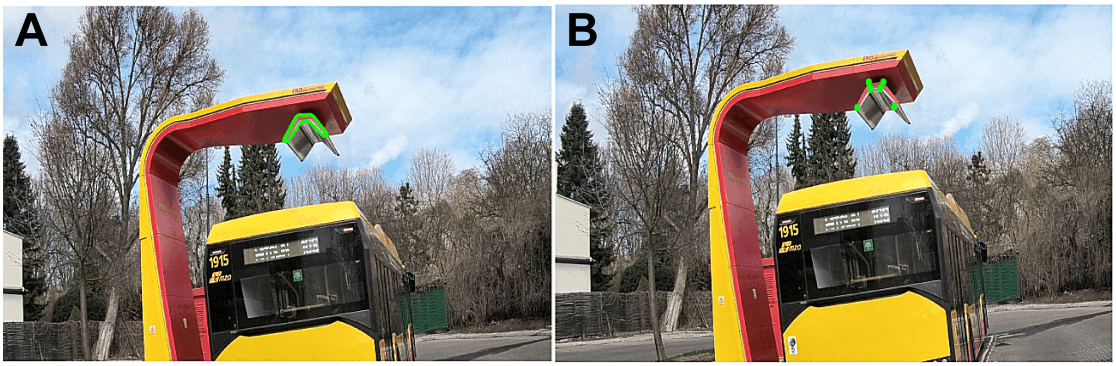

FIGURE 5.3: Examples of Mask R-CNN detection (A), and key points detection (B)

the localised points on the image and the projected points from the object model.

$$\mathbf{T}^* = \underset{\mathbf{T}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(\hat{\mathbf{p}}_i^{2d}, \pi\left(\mathbf{T}, \mathbf{p}_i^{3d}\right)\right)^T \left(\hat{\mathbf{p}}_i^{2d}, \pi\left(\mathbf{T}, \mathbf{p}_i^{3d}\right)\right), \tag{5.3}$$

where n is the number of points, $\hat{\mathbf{p}}_i^{2d}$ are the image coordinates of the i-th point of the charger detected on the image by the deep learning system, $\pi(.)$ is a camera projection function, $\mathbf{T}$ is a rigid transformation matrix (rotation and translation), and $\mathbf{p}_i^{3d}$ are coordinates of the i-th 3D object's point based on the 3D model of the charger.

However, this optimization-based method requires a good initial estimate of the camera pose. While this is not an issue when localising the bus along a predefined path toward the station, where previous pose estimates and odometry data can be utilised, the absence of an initial guess poses a challenge. To address this, a separate initialization procedure was developed. This involves running the solvePnP algorithm with several initial guesses within the maneuver's operational area and selecting the estimate with the smallest reprojection error. Consequently, this enhanced pose estimation system operates efficiently without requiring an initial pose estimate, thereby overcoming a common limitation of iterative solvePnP method.

5.3.1 Reprojection-based Pose Refinement

Keypoints defined on the object can be detected with varying levels of accuracy, influenced by several factors, including the camera viewpoint and the amount of motion blur present in the image. In such scenarios, it is possible for some keypoints to be accurately extracted while others are inaccurately positioned or even misplaced.

For setups involving a small number of keypoints (4-5), a sanity check procedure that incorporates geometric constraints during the neural network's inference process was implemented. This procedure refines the neural network predictions by addressing the same task described in equation 3.7.

Once the optimal transformation $\mathbf{T}^*$ is determined, the 3D coordinates of the keypoints are projected onto the image. Then the distances between the predicted points $\hat{\mathbf{p}}_i^{2d}$ and the projected

points $\tilde{\mathbf{p}}_i^{2d}$ are calculated. The maximum distance d_{max} is compared with the mean of the remaining distances d_{res}, which is multiplied by a parameter $\gamma_{ref} = 2$.

If the inequality $d_{max} > \gamma_{ref} d_{mean}$ is satisfied, this projection as the final prediction for the keypoints' locations is adopted. This condition ensures that only cases where there is a single point with an inaccurate prediction are refined, thereby improving the overall accuracy of the keypoint estimation.

This procedure was implemented in conjunction with the GAKN network for charger pose estimation using four keypoints. The reprojection-based refinement is executed during the post-processing stage, and as such, it does not affect the number of operations performed by the network or the number of parameters. This design choice ensures that the refinement process does not impose additional computational burdens on the network itself.

On average, the reprojection-based refinement step takes approximately 4 milliseconds per image. This processing time constitutes 11% of the total processing time for a heatmap resolution of 128×128 and only 8% for a heatmap resolution of 512×512. Therefore, incorporating reprojection-based refinement does not significantly increase the overall image processing time, maintaining efficiency across different resolutions.

By integrating the GAKN network with reprojection-based refinement, the issue of outlier keypoints is effectively addressed. This ensures that the system delivers more consistent and accurate pose estimations. The results of this approach are presented in section 6.4.

5.4 Pose estimation of vehicles with an unknown shape

5.4.1 Pose estimation from single image

In this section, the comprehensive pipeline designed for the pose estimation of a vehicle using a monocular camera is introduced, which operates without requiring explicit knowledge of the vehicle's 3D model. This approach is particularly significant as it circumvents the necessity of pre-existing 3D information, making it more versatile and broadly applicable. The entire processing pipeline for pose estimation is illustrated in Figure 5.4, providing an overview of the sequential steps and interactions between various components. Additionally, the details of the head architectures are depicted in Figure 5.5, offering deeper insights into their specific configurations.

The input to the pipeline is a cropped image that isolates the vehicle of interest. The pipeline is composed of several specialised modules: the 2D Keypoint Estimation Head, the 3D Keypoint Estimation Head, the Keypoint Score Head (KSH), and the Uncertainty Estimation Head (UEH).

The first module in our pipeline is a deep neural network designed to estimate the 2D coordinates of the vehicle's characteristic points on the image. Our methodology employs the HRNet48 architecture [107] as the backbone. This backbone is combined with a head dedicated to feature

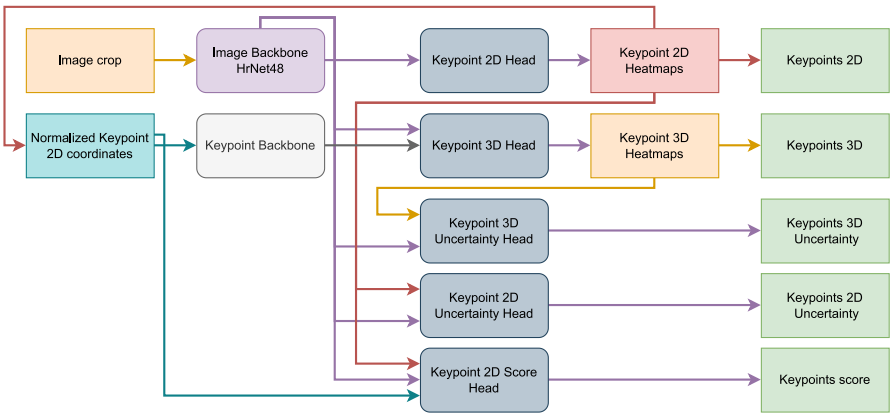

FIGURE 5.4: Architecture of the proposed vehicle pose estimation system

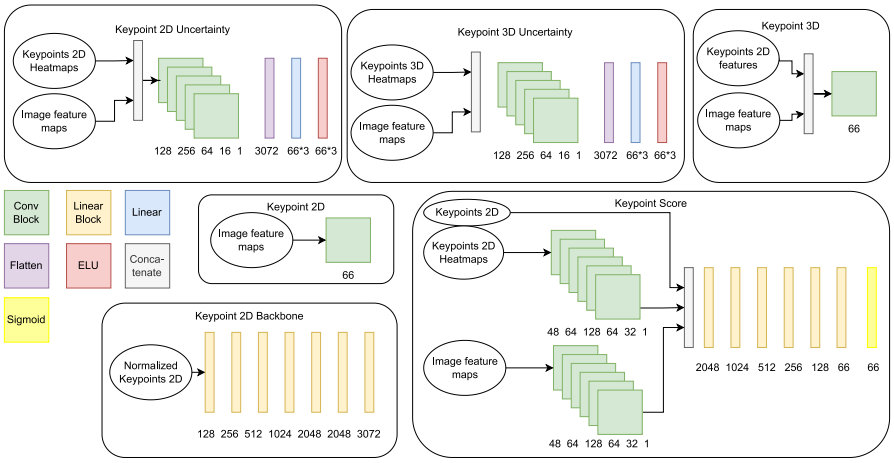

FIGURE 5.5: Architecture details of the implemented network heads. Conv Block means Convolution layer followed by batch normalization and ReLU activation. Linear Block means Linear layer followed by batch normalization and ReLU activation.

extraction and heatmap estimation for each point on the image, ensuring precise localization of the vehicle's keypoints. The second module predicts keypoints 3D as described in the Sec. 4.3.

The subsequent module - Keypoint Score Head in our pipeline is designed to assess the precision of the estimated points. Accurate localization of characteristic points is inherently challenging due to occlusions that occur from various viewing angles, which obscure a substantial portion of these points. Inaccurately estimated points can significantly distort the overall pose estimation, leading to unreliable results. To mitigate this issue, an additional head specifically for evaluating the accuracy of each estimated point has been developed.

It takes as input the image feature map, the estimated heatmaps, and the normalised coordinates of the estimated points. By processing these inputs, the KSH can assess the reliability of each point's estimation, selecting the points that are accurately estimated and distributed across the vehicle to ensure the best results pose estimate using the PnP algorithm.

The image feature maps and heatmaps undergo processing through a sequence of six convolutional layers, which results in the generation of two feature vectors, each with a length of 3072. These feature vectors, combined with the estimated point coordinates, are then concatenated

and further processed through a sequence of six linear layers. The final layer in this sequence is a sigmoid activation layer, which restricts the output values to a range between zero and one.

The output produced by this head consists of a set of N values, each representing the estimation precision score of a specific 2D point on the image. During the training phase, the learning targets are binary values: '1' for a correctly estimated point and '0' for an incorrectly estimated point. A point is classified as correctly estimated (assigned a '1' label) if the estimation error, normalised to the bounding box, is less than 0.04.

In addition, experiments were conducted using the keypoint 2D estimation errors normalised to the bounding box directly as learning targets for the KSH. However, the results obtained from this approach were less favorable compared to using binary targets derived from thresholding the error as described in the previous paragraph. The binary classification proved to be more effective for our models learning process.

The training process is organised into distinct stages, with each stage focusing on a specific aspect of the network. Initially, in the first stage, our attention is directed towards training the 2D and 3D point heads along with the backbone network. This foundational stage establishes the basic capabilities of the model in estimating both 2D and 3D points accurately.

After completing the initial stage of training, the best-performing model from this phase is selected. The selection is based on performance evaluation using two key metrics: the Percentage of Correct Keypoints (PCK) for 2D keypoints and the Mean Per Joint Position Error (MPJPE) for 3D point estimations.

The PCK metric is defined as:

$$PCK = \frac{n_{correct}}{n} \cdot 100, \tag{5.4}$$

where n is the total number of predicted points. For this metric, the number of correctly estimated points $n_{correct}$ is summed up. These points are defined as keypoints for which the error in coordinate estimation, when normalised to the bounding box, is less than 0.05.

The MPJPE metric is a measure used to evaluate the accuracy of our model's 3D point estimations. It is defined as the mean of the Euclidean distances between the estimated points and their corresponding ground truth points (Eq.(5.5)).

$$\text{MPJPE} = \frac{1}{n} \sum_{i=1}^{n} \left\| \hat{\mathbf{p}}_i^{3d} - \hat{\mathbf{p}}_i^{2d} \right\|_2 \tag{5.5}$$

where n is the total number of keypoints, $\mathbf{p}_i^{3d}$ represents the ground truth coordinates of the i-th keypoint in a 3D space and $\hat{\mathbf{p}}_i^{3d}$ represents the predicted coordinates of the i-th keypoint in a 3D space.

In addition, the training of the 2D and 3D keypoint heads incorporates a reprojection loss function. This function extends the approach presented in [76], which is designed to maintain the geometric consistency of the estimated coordinates. The reprojection loss function operates

by comparing the projected 3D points, transformed using ground truth data, with the predicted 2D points. This comparison ensures that the network's estimations of 2D and 3D points are not only internally consistent but also align accurately with real-world geometries.

This loss is defined as:

$$\mathcal{L}_{\text{reprojection}} = \sum_{i=1}^{n} \left(\left\| \pi \left(\mathbf{T}, \hat{\mathbf{p}}_i^{3d}, \mathbf{K} \right) - \hat{\mathbf{p}}_i^{2d} \right\|_2 \right)^2, \tag{5.6}$$

where π is the projection function, $\mathbf{T}$ is a ground truth pose, $\hat{\mathbf{p}}_i^{3d}$ are the estimated 3D coordinates of the i-th characteristic point, $\mathbf{K}$ is the camera intrinsics matrix, and $\hat{\mathbf{p}}_i^{2d}$ are the estimated 2D coordinates of the i-th keypoint on image.

The application of this reprojection loss function has been shown to enhance the accuracy of the point estimation heads. By ensuring geometric consistency between the 2D and 3D estimations, the loss function helps the network to produce more precise and reliable keypoint predictions.

After selecting the best-performing model from the initial training phase, the weights are frozen. This preserves the optimal results achieved for keypoint estimation, ensuring that the finely tuned parameters remain unchanged during subsequent training stages.

In the second stage of training, the focus shifts to the Uncertainty Estimation Head and the Keypoint Score Head. Training these components after the Keypoint 2D and 3D Heads allows us to leverage the well-tuned features provided by the backbone network and the point-generating heads. By building on this foundation, the UEH and KSH can be trained more effectively, enhancing the overall performance of the model.

The Uncertainty Estimation Head is trained to quantify the uncertainty associated with each keypoint estimation, providing valuable insights into the confidence level of the predictions. Meanwhile, the Keypoint Score Head is trained to evaluate the accuracy of the estimated keypoints, producing scores that reflect the precision of each point.

The fine-tuning of the UEH and KSH components is performed using the loss functions described by Eq. (5.7), (5.8)

$$\mathcal{L}_{stage1} = w_{repr} \cdot \mathcal{L}_{reprojection} + \mathcal{L}_{heatmap3Dxy} + \mathcal{L}_{heatmap3Dxz} + \mathcal{L}_{heatmap2D} \tag{5.7}$$

$$\mathcal{L}_{stage2} = \mathcal{L}_{uncertainty2D} + \mathcal{L}_{uncertainty3D} + \mathcal{L}_{keypoint\_score}, \tag{5.8}$$

where $w_{repr} = 1e^{-7}$ and $\mathcal{L}_{heatmap3Dxy}, \mathcal{L}_{heatmap3Dxz}, \mathcal{L}_{heatmap2D}, \mathcal{L}_{keypoint\_score}$ are defined as the Mean Squared Error loss function.

Given the 2D points in the image, their corresponding 3D points in the model, and the camera matrix $\mathbf{K}$, various PnP algorithms can be employed to derive the vehicle's pose. The network estimates both the 2D characteristic points of the car and the corresponding 3D points. In this research, two approaches were utilised: the EPNP algorithm [54] and a dedicated procedure called Solve-PnP-BFGS, implemented using the SciPy library [102]. To provide input for the PnP algorithm, a subset of N best-estimated points based on their KSH scores is selected.

The Solve-PnP-BFGS procedure operates by minimising the reprojection error using the BFGS optimization algorithm [11]. The reprojection error in this context is defined similarly to Eq. (5.6), but instead of the ground truth transformation $\mathbf{T}$, the optimised transformation is used. This optimization ensures that the estimated pose aligns as closely as possible with the observed data.

In the BFGS algorithm, the optimization search space is constrained by bounds applied to the estimated translation parameters. This constraint helps guide the optimization process within realistic limits, preventing it from exploring implausible solutions. The optimization is carried out five times, each time starting from a different randomly selected point within the search space. This multiple-start approach helps to mitigate the risk of the optimization process getting trapped in local minima. The final solution selected is the one that yields the lowest value of the cost function. The results of this approach are presented in section 6.5.3.

5.4.2 Propagation of point uncertainty to the pose uncertainty

The uncertainties estimated as described in section 3.4 for both 2D and 3D points can be propagated to determine the uncertainty of the estimated vehicle pose. To achieve this, the Unscented Transform (UT) method as described in [80] is used. The Unscented Transform is a method used to propagate probability distribution through nonlinear transformations. To propagate the uncertainty of an n-dimensional input it utilises a set of $2n+1$ sigma points (χ_i, $i = 0, ..., n$). These points are chosen to best represent the distribution's mean and covariance to capture the effects of nonlinearity without requiring derivatives or Jacobians. These sigma points are propagated through the nonlinear system, and the resulting set is used to compute a new mean and covariance, thereby reflecting the transformed distribution's properties.

The sigma points are calculated according to specific formulas (Eq. (5.9)):

$$\chi_0 = \mathbf{m_x},\ \chi_{2i-1} = \mathbf{m_x} + \sqrt{n+\lambda_\chi}\left[\sqrt{\mathbf{C_x}}\right],\ \chi_{2i} = \mathbf{m_x} - \sqrt{n+\lambda_\chi}\left[\sqrt{\mathbf{C_x}}\right], \tag{5.9}$$

for $i = 1, ..., n$, where $\mathbf{m_x}$ and $\mathbf{C_x}$ are the mean and variance of the estimated points, λ is a scaling factor calculated according to Eq. (5.10).

$$\lambda_\chi = \alpha_\chi^2(n + k_\chi) - n, \tag{5.10}$$

where α and k are the parameters influencing how far the sigma points are away from the mean.

By applying the PnP algorithm (a nonlinear transformation), a set of points is obtained from which the mean and covariance of the transformed points can be estimated. The reconstruction of the covariance matrix was done according to Eq. (5.11). In our implementation, the Unscented Transform parameters α_χ β_χ and k_χ are set to 0.9, 2, and 50, respectively.

$$C_y = \sum_{i=0}^{2n} w_i^c (Y_i - m_y)(Y_i - m_y)^T, \tag{5.11}$$

where Y_i are sigma points after passing through the nonlinear transformation, m_y is the mean of the transformed sigma points and w_i^c are the weights associated with each sigma point, used for calculating the covariance defined by Eq. (5.12, 5.13).

$$w_0^c = \frac{\lambda_\chi}{n + \lambda_\chi} + (1 - \alpha_\chi^2 + \beta_\chi) \tag{5.12}$$

$$w_i^c = \frac{1}{2(n + \lambda_\chi)} \quad \text{for} \quad i = 1, \ldots, n \tag{5.13}$$

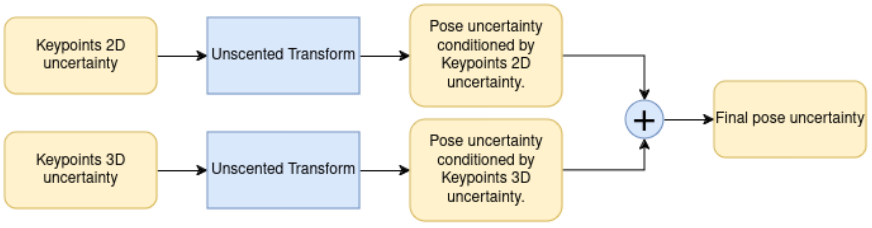

FIGURE 5.6: Block diagram of keypoints uncertainty propagation on the pose uncertainty. Yellow boxes are the covariance matrices.

The propagation process for 2D and 3D points is performed independently. Once the uncertainties have been propagated separately, the resulting covariance matrices, which have the same dimensions, are summed. This summation assumes statistical independence between the coordinates of the 3D and 2D keypoints (Fig. 5.6). The experimental results of this approach are presented in section 6.7.

Chapter 6

Applications of visual perception for autonomous vehicles

6.1 Introduction

In this chapter, the results of the experimental evaluation of the methods presented in the previous chapters will be discussed. The evaluation focuses on two specific scenarios: the docking maneuver of an electric bus to a charging station, and the estimation of vehicle poses in an urban environment. These practical applications facilitate the validation of the effectiveness and practicality of the proposed methods in real-world applications. The chapter will explore how each method performs under the unique challenges posed by these scenarios, providing insights into their potential for deployment in autonomous driving systems.

The integration of electric buses into urban public transportation systems is a growing trend, driven by the need for sustainable and safe transit solutions [8]. As these buses increasingly rely on electric charging stations mounted on pylons for recharging en route, the challenge of docking these long, articulated vehicles with precision becomes evident. This task, demanding in terms of spatial coordination and timing, requires considerable skill and experience from drivers. The complexity of maneuvering a large vehicle into a precise position necessitates technological interventions to ensure efficiency and safety.

Given this backdrop, there is a demand for an Advanced Driver Assistance System [48] tailored for electric buses. Such systems aim to support drivers, particularly those with less experience, by providing clear and actionable guidance for docking at charging stations.

The docking maneuver is defined as the positioning of a selected guidance point of the bus relative to the charging station's head, as depicted in (Fig. 6.1). While the initial position of the bus in the charger head's coordinate system is known, the positioning task is reduced to computing a feasible trajectory between the initial pose of the bus within the charging station's coordinates and the desired location of the pantograph tip.

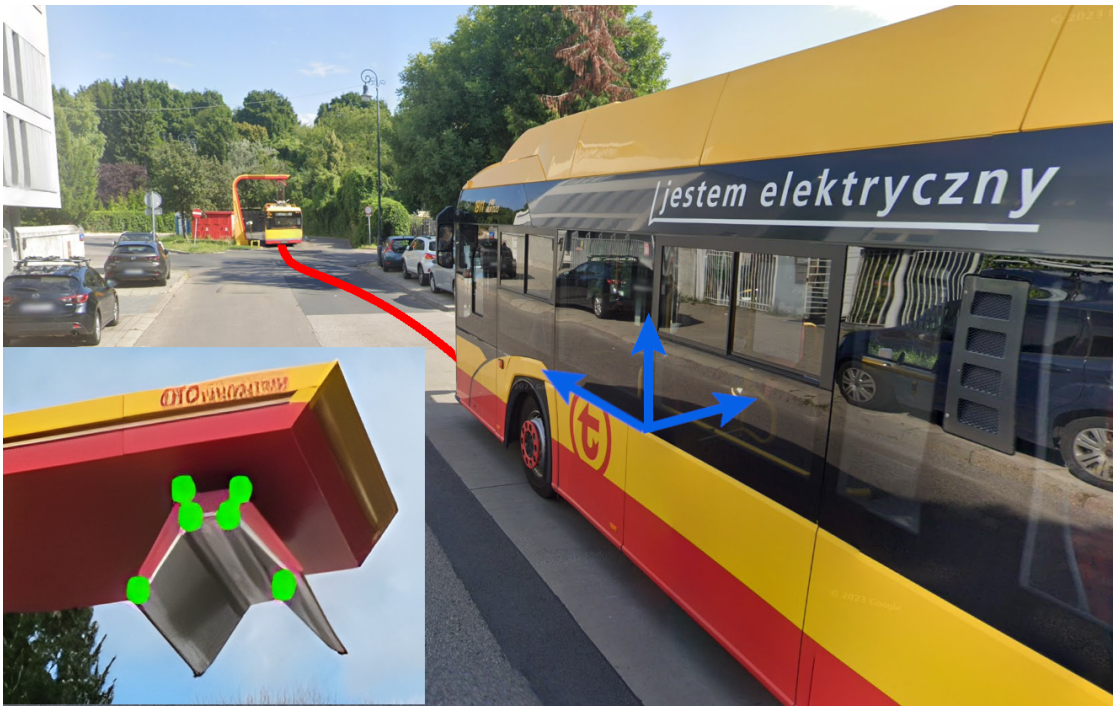

FIGURE 6.1: Illustration of the docking maneuver with an articulated bus. The visualised coordinate system's origin is coincident with the guidance point, while the red curve indicates the planned trajectory. The inset image shows a charging station with example salient features, as seen by the localization system's camera. Source: Google Maps

The process of docking involves several technical challenges, primarily the estimation of the bus's pose relative to the stationary charging station. Traditionally, this would involve technologies such as GPS. However, the standard GPS suffers from significant limitations in urban environments [119]. Signal outages and interferences commonly found in densely built areas make GPS unreliable for tasks requiring high precision. Although Differential GPS (DGPS) offers enhancements by correcting the GPS signal with additional real-time data from nearby reference stations, issues such as signal blockage by tall buildings, delayed corrections, and the necessity of a continuous network connection introduce complications that undermine its practicality for routine operations by bus fleets.

In light of these challenges, this research proposes an approach utilising passive vision technologies. Passive vision, leveraging cameras to capture the environment, offers several advantages over other sensor technologies. Cameras provide rich visual information that can be processed to extract detailed spatial data about the environment. When combined with advanced image processing and deep learning techniques, this data can be used to achieve accurate pose estimation necessary for precise docking maneuvers.

Specifically, a monocular camera setup is employed, a cost-effective solution compared to more complex stereo vision systems. This choice is dictated by the need for a system that is not only effective but also economical and easy to integrate into existing bus models. The monocular system, despite its simplicity, is capable of delivering the required performance through algorithms capable of interpreting the visual data in real-time.

Another practical application of the algorithms presented in previous chapters is the pose estimation of surrounding cars relative to the autonomous vehicle. This chapter presents an application of a pipeline for the pose estimation of surrounding cars with uncertainty quantification presented in Chapter 5.

The application of this pose estimation pipeline in autonomous driving is multifaceted. It enhances the vehicle's ability to make informed and safe decisions by providing reliable and precise data on the positions and movements of other vehicles [6]. Moreover, the inclusion of uncertainty measures aids in the development of robust decision-making algorithms that can effectively handle the unpredictabilities inherent in real-world driving scenarios.

The effectiveness of this pipeline is demonstrated through an evaluation using a real-world dataset. The system's performance is evaluated based on its accuracy, comparing it with current state-of-the-art systems in autonomous vehicle perception.

This chapter presents the practical applications of the algorithms detailed in Chapters 2 through 5, demonstrating their real-world viability through case studies and field implementations. An ablation study is conducted to systematically evaluate the impact of various components of the proposed methods, highlighting their individual and collective contributions to the system's overall performance. Furthermore, the chapter presents a comprehensive comparison of the obtained results with current state-of-the-art algorithms, showcasing the advancements and enhancements that this approach brings to the field of perception for autonomous driving.

6.2 Related work

The development of autonomous docking and charging systems for various types of vehicles has been an area of recent research. These systems aim to ensure that autonomous vehicles can efficiently recharge their batteries reducing human intervention, thereby enhancing their operational reliability and usability.

Luo [61] presented a foundational system involving a docking station equipped with an automatic recharging device for a security robot. This station utilises an artificial landmark to detect and recognise its presence, and a virtual spring model to guide the robot for accurate alignment and docking, demonstrating the feasibility of such systems through successful docking and recharging experiments. Building on the concept of autonomous docking, [96] explored a vision-only navigation and docking control system for an autonomous mower. This system employs a single camera and integrates the YOLO object detection framework with Double Deep Q-Networks for reinforcement learning, achieving centimeter-level accuracy in navigation from arbitrary starting points to the docking station. This showcases the potential of low-cost, vision-based solutions in enhancing autonomous operational accuracy without the need for external sensors. In urban environments, Clarembaux et al. [16] focused on the Furbot project, aiming to improve urban freight transportation using fully electric vehicles. The project enhances vehicle autonomy in docking by utilising onboard intelligent units that process LiDAR data to enhance perception and control during the docking processes, thereby optimising urban logistics operations.

Further addressing the challenges of electric vehicles, particularly regarding their short driving range and the inconvenience of manual charging, Miseikis et al. [66] introduced an automated robot-based charging station with 3D vision guidance. This system facilitates the accurate connection of chargers using shape-based matching methods and a three-step robot motion planning process, significantly simplifying the electric vehicle (EV) charging process. Similarly, Petrov [79] developed an innovative docking station architecture for electric vehicle recharging that features a hybrid control scheme for automatic docking. This scheme integrates an automated arm and an infrared beacon system for precise vehicle localization and docking, demonstrating enhanced control effectiveness through simulation and experimental results. Expanding the scope of autonomous docking technologies, Gong et al. [30] proposed a real-time planning and tracking control method for wireless charging of nonholonomic autonomous vehicles in open environments. This advanced method incorporates dynamic obstacle avoidance and precise posture control, significantly reducing docking errors to less than 3 cm, thus showcasing a substantial improvement over traditional SLAM-based planning and control algorithms.

In the field of autonomous driving, vehicle pose estimation is an important aspect that influences the performance of autonomous systems. This section reviews the existing literature on the topic, focusing on different methods developed for pose estimation. The following papers utilise the ApolloCar3D dataset which makes its application similar to the one discussed in this dissertation.

An approach presented in [34] addresses the challenge of view-invariant object detection and semantic keypoint pose estimation from a single RGB image, aiming to estimate the absolute pose of on-road vehicles using a monocular camera. This research introduces a deep hybrid architecture combining a Convolutional Neural Network and a Recurrent Neural Network (RNN) to enhance 6D pose inference. The proposed method leverages Long Short-Term Memory (LSTM) networks to filter out stationary vehicles and focus on moving ones. DeepMANTA [14] represents another approach, designed for simultaneous vehicle detection, part localization, visibility characterization, and 3D dimension estimation from a single image. The architecture employs a convolutional network and a coarse-to-fine object proposal mechanism to enhance vehicle detection. The outputs of this network feed into a real-time pose estimation algorithm, which determines vehicle orientation and 3D position. GSNet [38] is an end-to-end framework that jointly estimates 6DoF poses and reconstructs detailed 3D car shapes from urban street views. GSNet utilises a unique feature extraction and fusion scheme, which improves model performance. By implementing a divide-and-conquer strategy for 3D shape representation, GSNet achieves high detail in 3D vehicle reconstruction. The introduction of a multi-objective loss function enhances geometrical consistency and scene context, improving the accuracy of 6D pose estimation. Another study proposes the BAAM [52] algorithm for monocular 3D pose and shape reconstruction. BAAM reconstructs 3D object shapes by considering the relevance between detected objects and vehicle shape priors, followed by estimating 3D object poses using bi-contextual attention. This approach leverages inter-object relationships and scene context to improve pose accuracy. The study also introduces a 3D non-maximum suppression algorithm to eliminate spurious objects based on Bird-Eye-View distance. A learning-based framework for recovering vehicle pose in SO(3) from a single RGB image is introduced [56]. This approach extracts meaningful Intermediate Geometrical Representations (IGRs) to estimate vehicle orientation. The deep model

transforms perceived intensities to IGRs, which are then mapped to a 3D representation encoding object orientation in the camera coordinate system. A new loss function based on projective invariants allows the use of unlabeled data during training, enhancing representation learning.

6.3 Object detection

This section presents the results of the experimental evaluation of the method for attention visualization of object detection network and guided learning procedure presented in chapter 2.

6.3.1 Experimental verification of attention visualization

To train and evaluate an electric bus charger detector two datasets were prepared. One dataset was collected using a real charger placed at the campus of Poznan University of Technology (PUT), as shown in Fig. 6.2 A. While this setup allows for gathering new recordings when needed, it does not fully replicate real-life scenarios due to the fixed background. In this controlled environment, 2000 images, referred to as the PUT dataset, were extracted from video sequences.

To evaluate the detection system's performance in real-world conditions, a second dataset was collected at the Solaris Bus & Coach (SBC) [17] production facility. This dataset includes images taken from the top of a bus maneuvering towards an electric charger mounted on a pylon, simulating typical operational conditions. In these settings, another 2000 images were captured from various angles and distances as depicted in Fig.6.2 B.

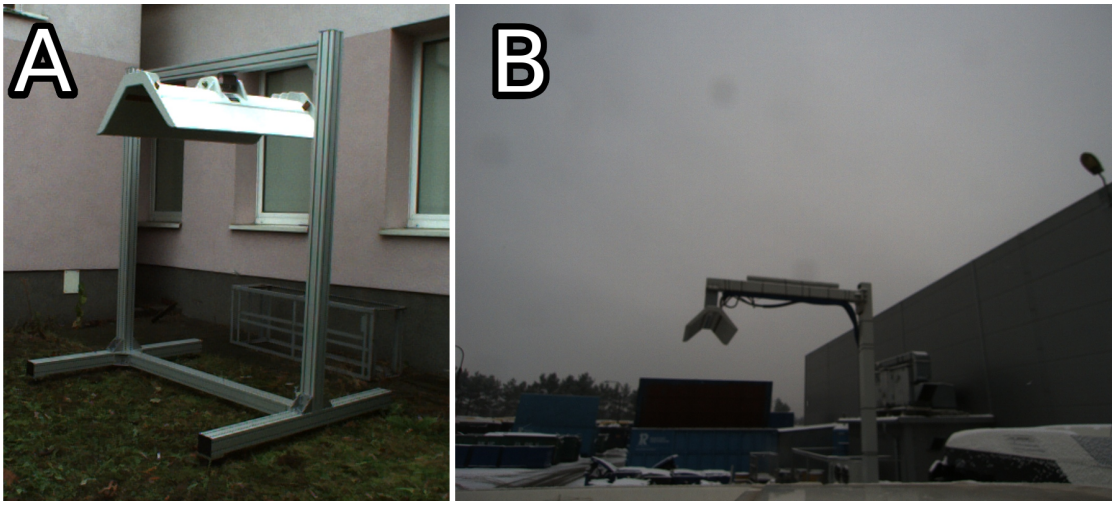

FIGURE 6.2: The exemplary images of the electric charger taken on a setup at PUT (A) and the Solaris Bus & Coach (B)

A baseline model was trained using a base dataset comprising 2000 images from the PUT dataset and 1000 images from the SBC dataset.

An example of attention visualisations is presented in Fig. 6.3. Regions marked by the warmer colors show where the network's attention is focused. Fig. 6.3 A demonstrates that the warmer

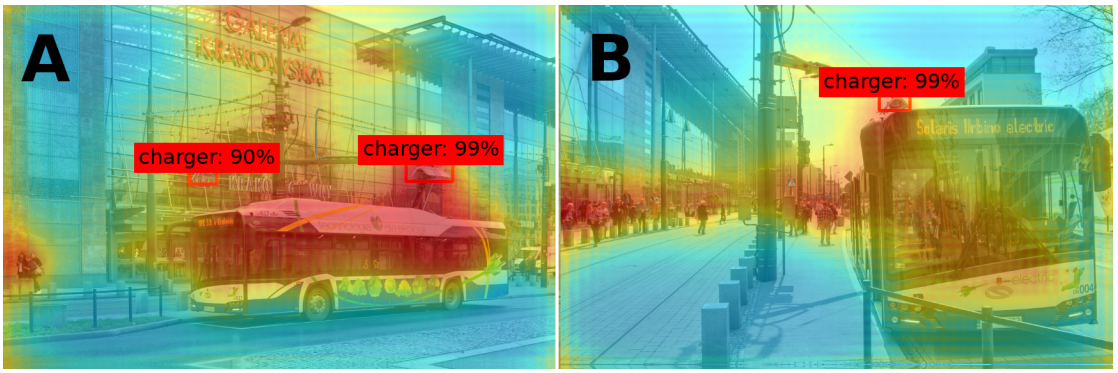

FIGURE 6.3: Visualization of attention maps for a model trained on the base dataset.

regions of the heat map directly correspond to locations of false positive detections, specifically on the glass facade of the building and the bus.

In a more complex scenario illustrated in Fig. 6.3 B, the electric charging post is accurately identified, yet a closer examination of the heat map indicates that despite generating the correct bounding box, the network's focus was significantly influenced by the presence of a crowd on the street and a bus occupying a substantial portion of the image. This instance of utilising this method reveals an important insight: the network, having been trained on a relatively simple dataset, may not perform as anticipated under conditions akin to those depicted. The presence of minor variations, such as an increase in the number of people within the frame, has the potential to misguide the network's focus towards incorrect areas of the image. This suggests that while the network can correctly identify specific objects, its ability to maintain focus on relevant areas can be compromised in complex, dynamic environments, highlighting the need for training on more diverse datasets to enhance the model's robustness in real-world scenarios.

The examination of the scenarios previously discussed uncovers that the system's accuracy could be enhanced by incorporating training images that mainly feature buses, cars, and pedestrians. These elements are often incorrectly identified as electric chargers by the system. This misidentification, especially concerning people, might seem paradoxical at first glance. However, it is important to acknowledge the operational mechanics of CNNs, which analyse a conglomeration of local features to make identifications. Through this lens, it becomes apparent that numerous minor similarities, which typically escape human notice, can collectively lead to erroneous recognition by the system. This insight underscores the need for expanding the training dataset to include a wide variety of real-world elements. By doing so, the model is better equipped to discern between genuinely relevant features and coincidental resemblances, thereby reducing the likelihood of false identifications and substantially improving system performance in diverse environments.

6.3.2 Experimental verification of guided learning procedure

Experiments concerning guided learning procedure were conducted using a base training dataset (SBC_50) comprising 2000 images from the PUT dataset and 1000 images from the SBC dataset,

supplemented with additional images that did not contain any electric chargers (negative examples). The proposed configurations are presented in Table 6.1. Learning results obtained for these configurations are presented in Fig. 6.4.

Image source	Dataset			
	SBC_50	BUS_25	BUS_100	BUS_500
PUT	2000	2000	2000	2000
SBC	1000	1000	1000	1000
Buses, cars	-	25	100	500
People	-	25	100	500

TABLE 6.1: Sequences used for system training when additional images with buses and people are used

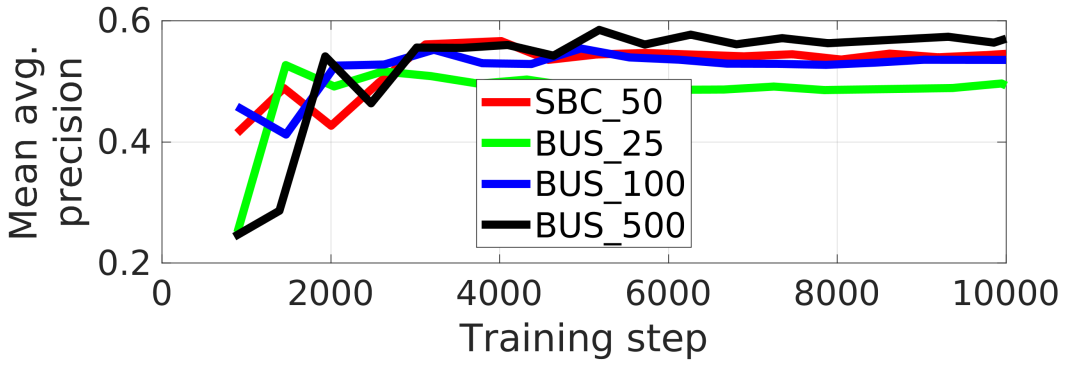

FIGURE 6.4: Learning curves obtained when the base dataset (SBC_50) system was augmented with additional samples containing buses and crowded places without additional positive examples of electric chargers

The augmentation of the dataset with additional object examples guided by the attention heatmaps acquired using a method introduced in Section 2.3.1 has improved the results. This approach has reduced false positive detections caused by buses and glazed panels, enabling a single, accurate detection (Fig. 6.5A). In Fig. 6.5B, the electric charger is correctly recognised, and the network's attention is no longer distracted by crowd or bus elements, indicating increased robustness to local background variations.

FIGURE 6.5: Visualization of attention maps for a model trained on a BUS_500 dataset extended with negative examples. Compare with Fig. 6.3

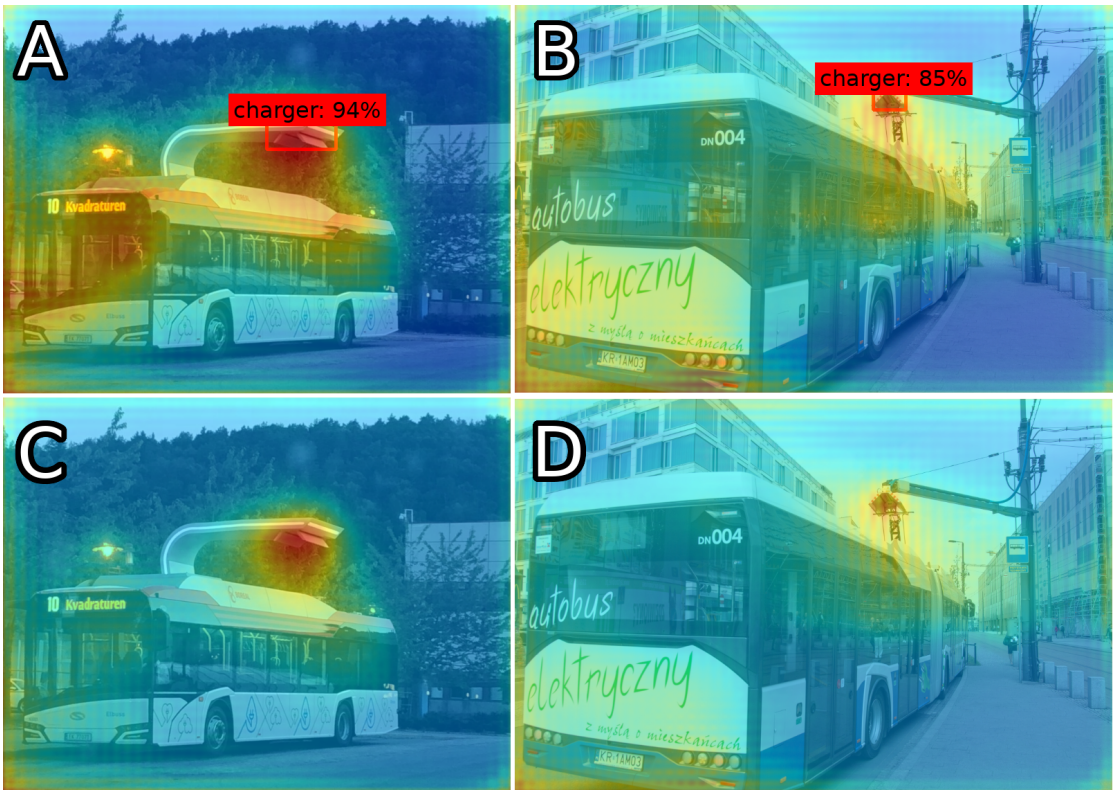

FIGURE 6.6: No detection examples when the network was trained on the BUS_500 dataset (C, D). For these examples the electric chargers were correctly recognised when trained on BUS_25 (A, B)

Surprisingly, adding only a small number of additional negative examples generally leads to worse performance compared to not including these images at all. The limited amount of diverse training data "confuses" the network, resulting in decreased performance. Additionally, providing a large number of negative examples does not necessarily improve performance, as demonstrated in Fig. 6.6. In these cases, the network trained on the BUS_500 dataset failed to produce any correct bounding boxes, even though the heatmaps indicated that the system focused on the correct areas of the images. Hence, it is needed to balance the amount of additional augmentation.

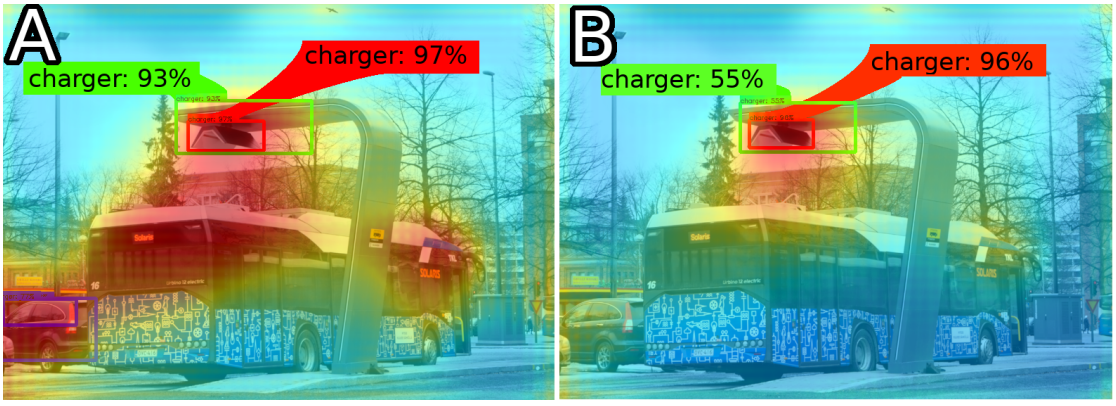

FIGURE 6.7: Electric charger detections marked on heat maps when the system was trained on the base (A) and an augmented dataset (B)

However, guided augmentation increases the likelihood of correctly detecting bounding boxes for the object of interest while reducing the probability of incorrect detections. This is demonstrated in Fig. 6.7, where the charger is detected twice with bounding boxes of different scales. Selecting the correct bounding box was challenging when using the base dataset for training, as the detection probabilities were nearly equal (93% and 95% – Fig.6.7A). In contrast, training with the augmented dataset significantly increased the network's confidence in the correct detection (96%) compared to the rough one (55%), as shown in Fig. 6.7B. This improvement allows for selecting the correct bounding box using simple thresholding.

In real-world settings, electric charging stations are not often prominently featured in most photographs. Based on this observation, the efficiency of the proposed solution was evaluated using the "00" sequence from the KITTI dataset, which includes a total of 4541 images without any bus electric charger imagery. When training with the base dataset, which comprised a mix of cars and people, a substantial number of 8487 false positives were encountered, with various cars and individuals mistakenly identified as charging stations, as shown in Fig. 6.8 A. The original image displayed only one misidentification; however, the corresponding attention heat map revealed that the network's focus was excessively concentrated on all automobiles present. This result was somewhat anticipated given the lack of cars in the utilised training data.

Remarkably, when 1000 new pictures of buses and people were added to the training dataset, the number of mistakes the system made dropped to just 207. This decrease to 2.4% of the initial count of false positives underscores the system's enhanced performance, as shown in the attention heat maps post-guided learning (Fig. 6.8 B), showing a reduced tendency to incorrectly recognise objects that were previously mistaken. Thus, despite the initial training dataset not closely mirroring the testing environment, the integration of additional, targeted imagery based on insights from the attention visualization algorithm improved the system's performance. The final number of false positives, given the system's initial unfamiliarity with the testing environment's specific conditions, is deemed to be within an acceptable range. This outcome demonstrates the potential of leveraging guided learning and targeted data augmentation to refine and improve object detection systems, even in the absence of directly analogous training materials.

FIGURE 6.8: A) The false detection reported for the KITTI sequence 00 when trained on base. B) The system trained on BUS_500 is no longer confusing cars with electric charger.

6.4 Vechicle pose estimation in assisted bus charging

In this chapter, the results of evaluating the pose estimation methods for electric bus chargers presented in section 5.3 will be discussed. The first part compares RKN and MRHKN networks and presents the ablation study of the MRHKN network, and the second part presents an evaluation of the GAKN network.

Docking a bus to a charging station's head necessitates accurate localization within the station's coordinate system over a range of distances, beginning from nearly 40 meters. Ensuring precise results at the initial stages of this maneuver is difficult due to the small size of observed objects, while at the final stages, the charging station may not fit entirely within the image frame. Additionally, the lateral distance offset between the pantograph's tip and the charger's head, as well as the angular offset between the charger's head's longitudinal axis and the approach direction, must be minimal by the end of the maneuver to avoid mechanical damage. However, some tolerance in both translational and rotational components of the estimated pose is allowed, given that the charger's head's mechanical design accommodates lateral offsets and allows safe docking with slight angular inaccuracies [89].

The pose estimated by the vision system is not directly employed in the path planning and steering of the ADAS. Instead, it is combined with an odometric pose estimate derived from a mathematical model of the vehicle and measurements from the bus's proprioceptive sensors. This method addresses occasional lapses in pose estimation caused by occlusions or image artifacts,

such as those from direct sunlight, and enables the provision of pose estimates to the control system at a higher frequency. However, bus odometry and its integration are part of the ADAS control system, which is beyond the scope of these experiments. The focus is on the performance of vision-based localization, without using odometric data to enhance the pose estimates in the presented experiments.

Due to the extensive range of observation distances, a high-resolution FLIR Blackfly S camera is utilised (5472×3648 pixels). This high resolution ensures optimal performance, which can be scaled down to balance performance with the cost for real-world applications (see Sec. 6.4.6). The camera's field of view (FoV) also plays an important role. A smaller FoV enlarges the object in the image but limits the range of possible maneuvers that keep the charger within the camera's view. Charging stations are sometimes located in bus bays that require rather sharp steering during docking. Thus, it was assumed that a 60° FoV is suitable for all realistic scenarios, providing the necessary resolution for subsequent processing. Considering these design choices and the technical requirements of the bus manufacturer, the goal is to create a vision-based localization system with a translational error under 0.35 meters and a rotational error under 1°.

6.4.1 Experimental setup and image sequences

The dataset used for training the neural networks was collected during May and June, primarily in sunny weather conditions. Two electric buses were utilised for data collection: a 12-meter long single-body bus and an 18-meter long articulated bus. The bus driver executed various paths towards the charging station to create a diverse dataset. The training dataset comprises 1000 manually labeled images, which were augmented by applying random changes in brightness and contrast, as well as random resizing and cropping. This augmentation process expanded the training dataset to a total of 10,000 different samples.

The proposed methods were evaluated using a dataset of images collected over five days in late autumn with an 18-meter articulated bus, under various weather conditions including cloudy, rainy, and sunny (Fig. 6.9). This dataset comprises 81 sequences in which the bus followed different trajectories toward the charging station. Data diversity was achieved by starting from different points and orientations, navigating along curved or slalom-like paths, and varying the bus speeds along these routes (Fig. 6.10). Throughout these maneuvers, the vision-based positioning system was active, though the bus driver did not utilise the ADAS-generated driving suggestions.

The driver was intentionally instructed to perform maneuvers that deviated from the typical approach to the charging station to create a more varied set of trajectories, including oscillations and sharp turns. These maneuvers resulted in numerous trajectories that did not conclude with successful docking (indicated in red in Fig. 6.10), aiming to test if the vision system could still position the bus accurately in such scenarios. In this dataset, the total duration when the entire charging station's mast is visible on camera amounts to 1630 seconds (approximately 27 minutes), equating to 12,366 frames. Due to its smaller dimensions, the charging station's head remained visible for a longer period, totaling 1783 seconds (around 30 minutes), or 13,530 frames.

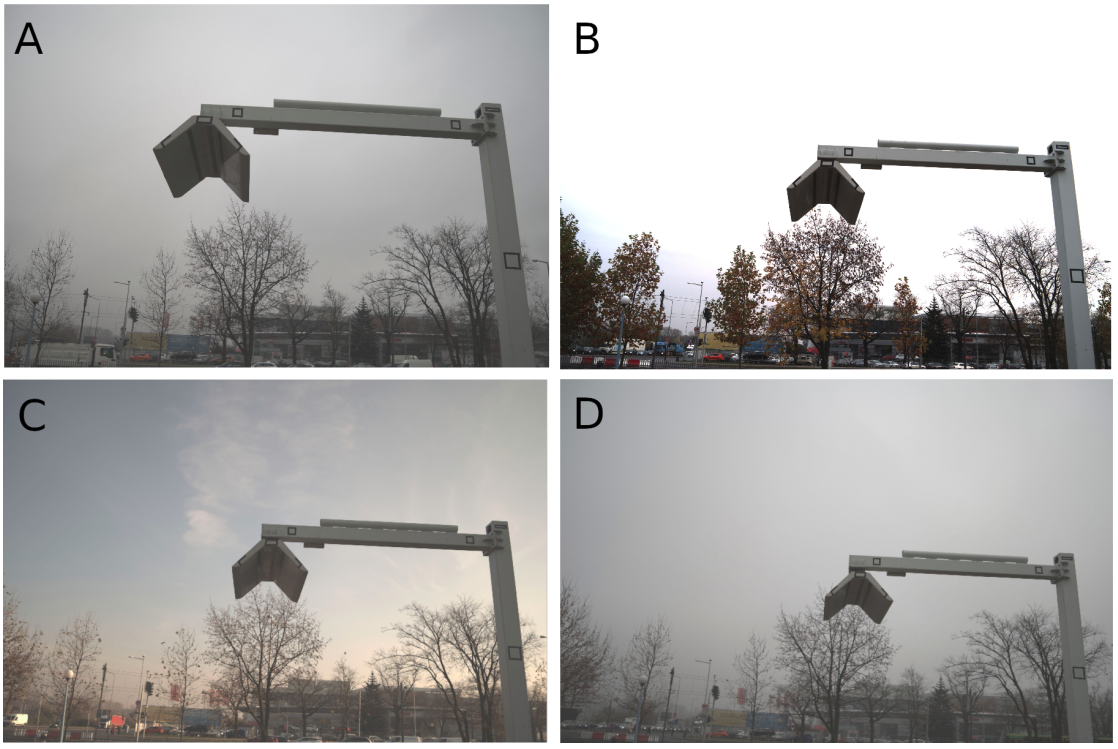

FIGURE 6.9: Example images from the test dataset with different weather and lighting conditions - cloudy morning (A), sunny midday (B), sunny morning (C), foggy morning (D).

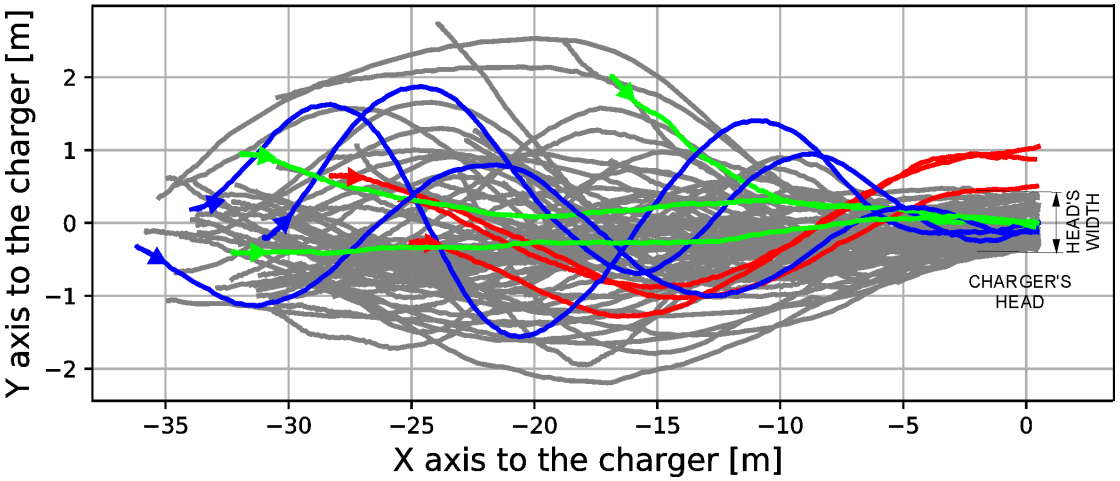

FIGURE 6.10: Bus trajectories used to gather the evaluation sequences. Colored lines highlight a few representative trajectories that are straight and end in proper docking (green), are unrealistic in real-world scenarios, but still end in proper docking (blue), or miss the charger's head by a large margin imitating a driver not following the suggestions of ADAS (red). Short arrows demonstrate start orientations for example trajectories. Notice that the vertical axis is scaled differently than the horizontal one in order to make the plot with a large number of trajectories more readable.

No publicly available dataset could be found to evaluate this approach on third-party data. Specifically, human pose estimation datasets, such as MPII Human Pose [3], cannot be utilised for a fair comparison of keypoint detection, as this system is specifically designed for positioning relative to a rigid object.

6.4.2 Ground truth and evaluation procedure

To evaluate the localization method's performance, a DGPS system (Ublox C099-F9P boards with ZED-F9P modules) was employed, featuring two receivers mounted on the bus and one external reference station placed near the experimental site. The DGPS system operated in a moving base scenario, utilising external corrections from the reference station to achieve an approximate accuracy of 1 cm in position and 1 degree in orientation while functioning in RTK (Real-Time Kinematic) mode [1].

The accuracy of the proposed camera-based system was assessed by comparing its estimates to those provided by the DGPS system, considering the requirements for motion planning and control. In practice, some detections from the neural networks may be erroneous and need to be filtered out. To achieve this, the system checks the alignment between the detected keypoints and the 3D model points projected onto the image plane, rejecting measurements if the Root Mean Squared Error (RMSE) of all charger points exceeds 10 pixels. The detections with smaller RMSE errors are considered valid (accepted detection) and are then evaluated by comparing the 2D pose of the camera (location on the ground plane and orientation as a single yaw angle) to the

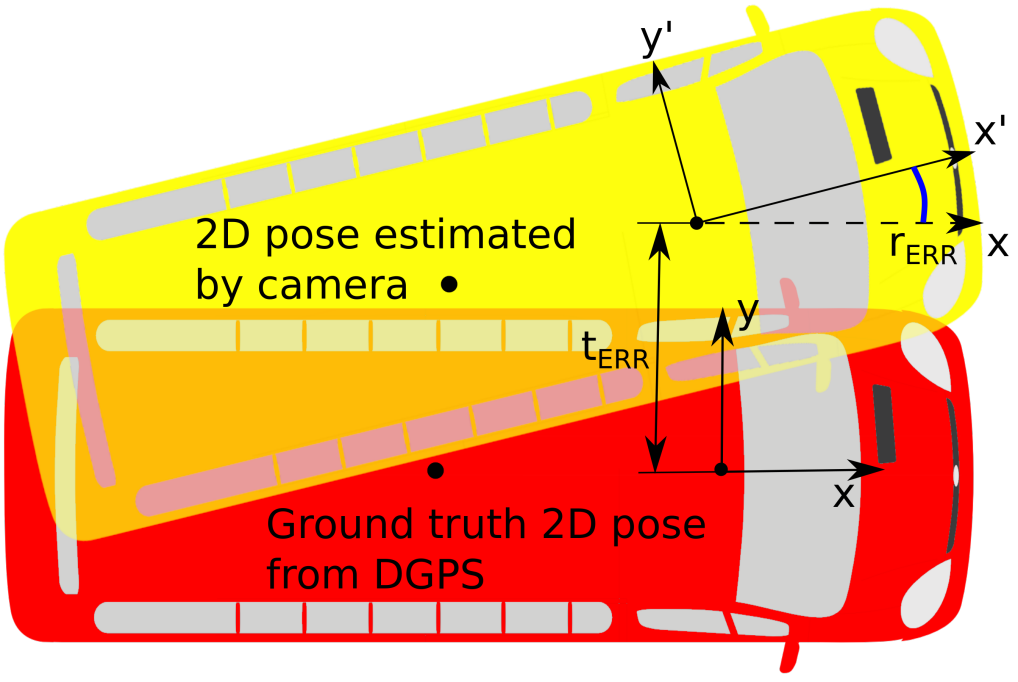

FIGURE 6.11: The vision-based localization system is evaluated with respect to the translation error on the ground plane (2D position) and the orientation error (yaw) understood as a difference between the estimated and the ground truth heading of the bus.

DGPS measurements. This evaluation ensures that the errors in the 2D pose accurately reflect the components of the bus pose that are necessary for motion planning and control procedures (Fig. 6.11).

Despite rigorous efforts, a small percentage of the computed camera poses may display significant translational errors. These errors can be easily rejected during path execution by considering the temporal relationships between consecutive detections. Nevertheless, each detection was treated independently, and these inaccurate detections were included in the evaluation and statistics. This approach was chosen because the focus is solely on evaluating the vision-based localization method, without integrating the bus odometry model, which must be separately identified for each vehicle type [65].

For each evaluated configuration of the positioning system, plots of the cumulative distribution function (CDF) of errors are presented. These plots visually distinguish the number of valid detections and illustrate the entire distribution of errors reported on the testing sequences. In the experiments, multiple initial guesses were used for each detection to ensure independence from previous ones. Odometry was not utilised, and DGPS was employed solely to obtain ground truth data.

6.4.3 Proposed processing pipeline

High-resolution images are utilised, which cannot be processed in real-time with standard hardware and neural network architectures. Given that the charging station's mast occupies only a small portion of the image from long distances, a two-stage processing pipeline was implemented. This pipeline initially detects the object of interest and subsequently identifies the keypoints belonging to that object (Fig. 6.12).

In the first step, the frame is resized to 960×960 pixels and processed by the Faster R-CNN network to detect the charging station. Images at this resolution are sufficient for accurately detecting charging stations in the given scenarios. Once the object detector network provides the bounding box coordinates, the region of interest is cropped from the original high-resolution image (Fig. 6.13). This region of interest (ROI) is then resized to 960×960 pixels and processed by another neural network to determine the keypoint positions on the object. This approach allows us to employ a common object detector architecture on high-resolution images from the camera and utilise the ROI at the maximum possible resolution, ensuring the best keypoint estimation accuracy.

Consequently, the camera's pose can be estimated using an algorithm designed to solve the PnP problem as described in Section 5.3.

The presented approach also necessitates detailed 3D locations of the keypoints on the charging station. Although a CAD model can serve this purpose, the experiments involved a mockup of the charging station that was partially assembled using non-standard elements. To accommodate this, a detailed 3D model of the station was obtained using a SURPHASER 100HSX 3D laser scanner, which captured a mesh-based model from a single viewpoint with an accuracy of 1 mm. This method allowed us to adjust the location of points as needed, even after the mast was fully

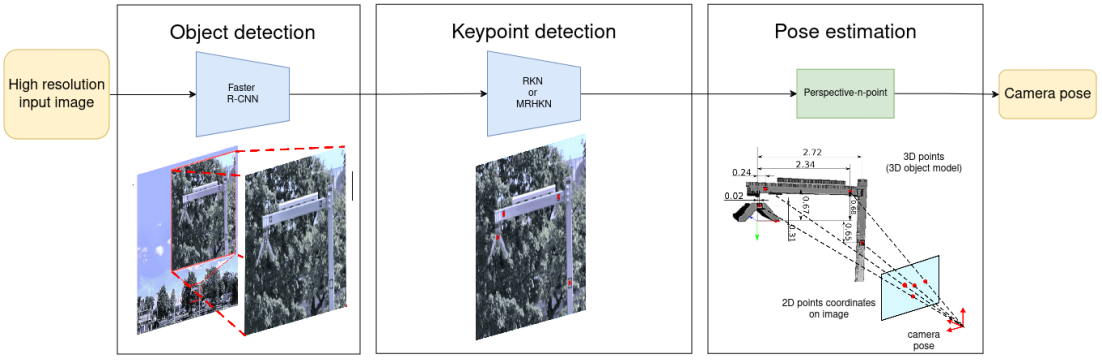

FIGURE 6.12: The block diagram illustrates the image processing and pose estimation pipeline employed for positioning electric buses relative to a charging station. The CAD model visualises the 3D mesh acquired from the SURPHASER 100HSX, complete with annotated dimensions between markers.

FIGURE 6.13: Visualization of the two-step processing pipeline: The object detector first analyses the full input frame at a reduced size and predicts the position of the charging station on the image (the small red rectangle). Subsequently, the ROI containing the charging station is cropped from the full-resolution image for further processing (the large red frame).

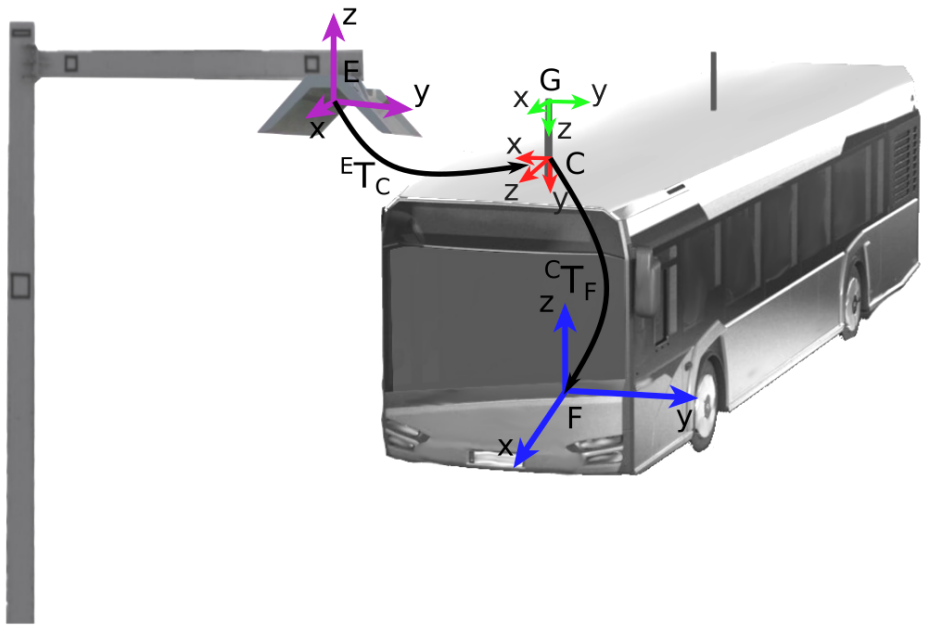

FIGURE 6.14: Overview of the coordinate systems: camera (C), DGPS (G), charger's head (E), and bus front axis (F).

mounted. For system deployment, these markings are envisioned to be pre-mounted on the mast components before installation at the desired location. Consequently, a production-ready system would utilise the 3D CAD model, obviating the need for an accurate 3D laser scanner.

The end result of this processing pipeline is the estimation of the pose of the charging station (E) relative to the camera coordinate system (C), denoted as ${}^{C}\mathbf{T}_{E}$ (Fig. 6.14). However, in practical applications, the system must provide the pose of the bus's front axis (F) relative to the charger coordinate system (E). This information is required for planning and controlling the motion of the bus as it approaches and docks with the charging station. To accurately evaluate the system and gather ground truth data, two masts were mounted to the roof of the bus.

In the first setup, a FLIR camera and a GPS antenna were mounted at the front of the bus. The second mast, which was positioned approximately 5 meters behind the front setup, supported another GPS antenna. This configuration was designed to achieve an accurate orientation estimate using DGPS. The final estimate is computed as:

$$ {}^{E}\mathbf{T}_{F} = \left({}^{C}\mathbf{T}_{E}^{-1}\right) {}^{C}\mathbf{T}_{F}, \tag{6.1} $$

where ${}^{E}\mathbf{T}_{F}$ is the pose of the bus's front axis with respect to the charger, ${}^{C}\mathbf{T}_{E}$ is the original measurement of the electric charger in the camera coordinate system, and ${}^{C}\mathbf{T}_{F}$ is the pose of the camera coordinate system with respect to the front axis of the bus.

The camera's position on the bus (i.e. the ${}^{C}\mathbf{T}_{F}$ transformation) was established using CAD files. This position was then verified through manual distance measurements and the attitude data of the camera setup, which was obtained from an XSens MTi IMU attached to the camera's mast.

FIGURE 6.15: Location of keypoints for charger pose estimation: four points located on the head and called *head* (A), four points located on the natural corners of the head and the mast called *corners* (B), and four points located inside the artificial markers called *markers* (C).

6.4.4 Selection of characteristic points

Two deep neural network architectures presented in section 3 (the RKN and the MRHKN) have been tested across three different spatial arrangements of keypoints.

The first arrangement, referred to as the *head*, consists of four keypoints located at the corners of the charger's head (Fig. 6.15 A). The second configuration, known as *corners*, uses two keypoints at the corners of the head and two additional keypoints on the supporting mast where the head is attached (Fig. 6.15 B). The third configuration, termed *markers*, involves keypoints placed within simple artificial landmarkssmall rectangles made of black tape positioned on both the head and the mast (Fig. 6.15 C).

Evaluating both neural network architectures allowed to compare their performance. By testing different spatial arrangements of keypoints, the study aimed to examine the impact of spatial layout and the type of physical features (i.e. natural corners or markers) on both the recall rate of the point detector and the accuracy of the computed pose. These findings help identify the optimal configuration of keypoints on the charging station for the best performance.

The initial aspect of detection efficiency to be compared when evaluating RKN is the ratio of accepted detections to the total number of frames where the charger is visible. The accepted detections are the detections where the RMSE compared to the 3D model points projected onto the image plane is less than 10 px. The *head* configuration performed the worst, correctly detecting keypoints in approximately 77% of frames. In contrast, the *markers* and *corners* configurations demonstrated improved performance, achieving coverage rates of 84% and 89%, respectively.

The pose error evaluation results for this approach are presented in Fig. 6.16. One notable observation is the almost linear distribution of translation errors. The slopes of the *markers* and *corners* configurations are similar and steeper compared to the *head* configuration. The distribution of rotation errors for the *markers* and *corners* configurations is more convex compared to the *head* configuration, indicating significantly better performance in estimating rotation angles.

Quantitative analysis shows that the median 2D translation error for the *head* configuration is nearly 4 meters, with a median yaw angle error of 10 degrees. These errors can be attributed to the dense packing of keypoints in the image. As described in the RKN approach, the method's

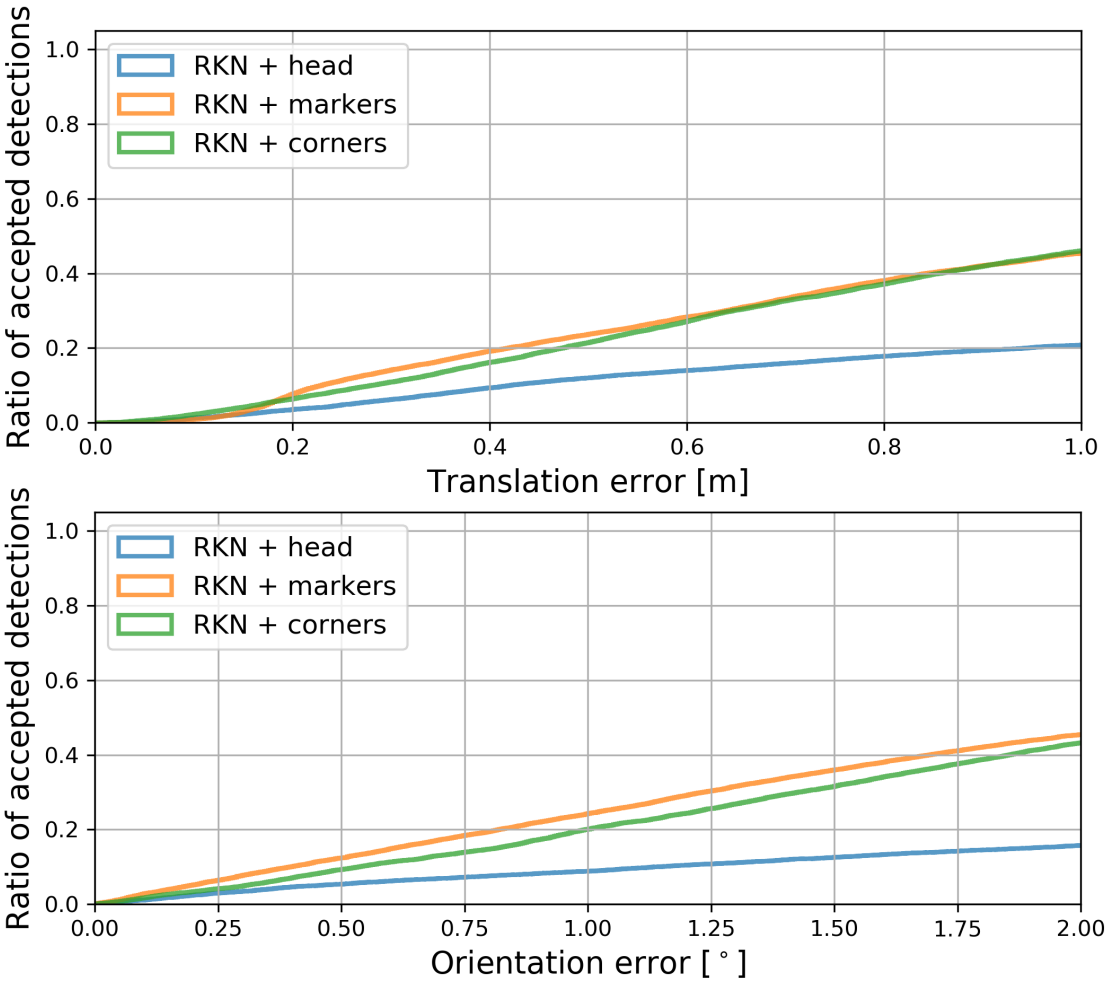

FIGURE 6.16: Cumulative distribution functions of 2D translation error (up) and orientation error (down) for the considered scenarios using RKN with *head*, *markers*, and *corners* points for pose estimation.

32×32 pixel bottleneck likely causes a loss of information about the locations of densely packed points during image processing.

The dense packing of keypoints also leads to unreliable camera pose estimates from the solvePnP algorithm, as small changes in keypoint locations result in relatively large changes in the estimated 3D pose. Additionally, some keypoints may have been inaccurately labeled during training due to the rounded corners of the charging station's head. Furthermore, the significant variations in the range of object observation throughout the maneuver make it challenging to maintain proper camera focus for all frames, resulting in some blurry images (Fig. 6.17 A).

The *corners* and *markers* configurations achieved median 2D errors of 0.97 meters and 1.94 degrees, and 0.91 meters and 1.96 degrees, respectively. These errors are significantly smaller than those observed with the *head* configuration and are quite similar to each other, making it difficult to determine which is superior. However, despite the improved performance, neither the *corners* nor the *markers* configuration meets the accuracy requirements for localization in the ADAS system.

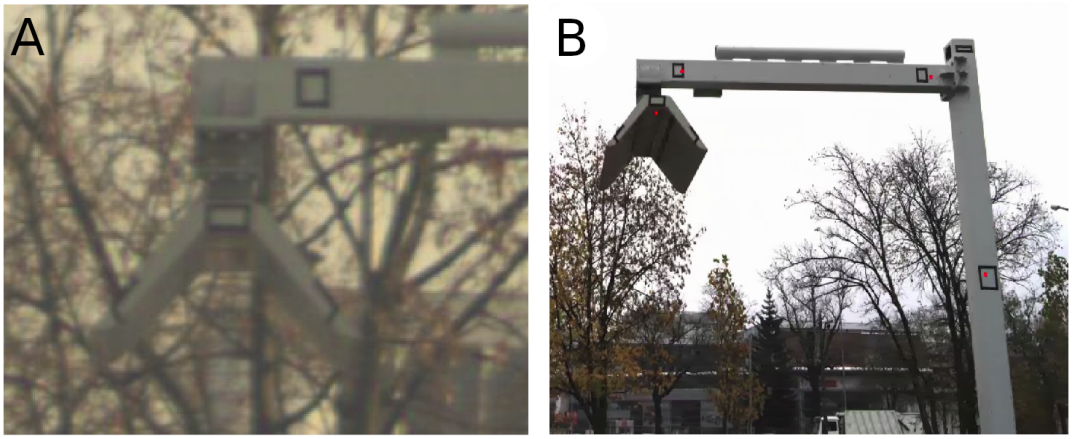

FIGURE 6.17: Example of a blurry image of the head with round corners (A) and inaccurately detected keypoints from RKN (B).

A closer examination of the results revealed inaccuracies in the estimated keypoint locations for both configurations. For example, the keypoints often missed the centers of the landmarks significantly, as illustrated in Fig. 6.17 B. This indicates the need to modify the image processing system to achieve more precise keypoint localization.

The MRHKN approach was evaluated with three distinct point configurations on the same dataset. The localization errors are shown as cumulative error distributions in Fig. 6.18.

Evaluation of the *MRHKN + head* version resulted in 87% of accepted detections, with a median 2D translation error of 6.46 meters and a median yaw angle error of 18°. These results indicate that this version is unsuitable for ADAS localization.

Although the *MRHKN + corners* version had only 66.6% accepted detections, the error curves in Fig. 6.18 demonstrate an improvement in pose estimation accuracy compared to the RKN model. The median 2D translation error of 0.28 meters and the median yaw error of approximately 0.6 degrees underscore the importance of keypoint locations in pose estimation. Most invalid detections occurred at distances greater than 25 meters, possibly because this method uses two points on the charger's head, which are not clearly visible from long distances, as previously discussed.

The best result, with 90.9% of accepted detections, was achieved by the *MRHKN + markers* version. This configuration appears sufficient for localization purposes. The quantitative evaluation showed a median 2D translation error of 0.17 meters and a median yaw error of 0.41°. This version demonstrates better keypoint detection from long distances, with significantly smaller median errors for camera positions more than 25 meters away from the charging station compared to the *MRHKN + corners* version. These results confirm that even simple and inexpensive artificial markers can enhance the robustness of the keypoint detection process.

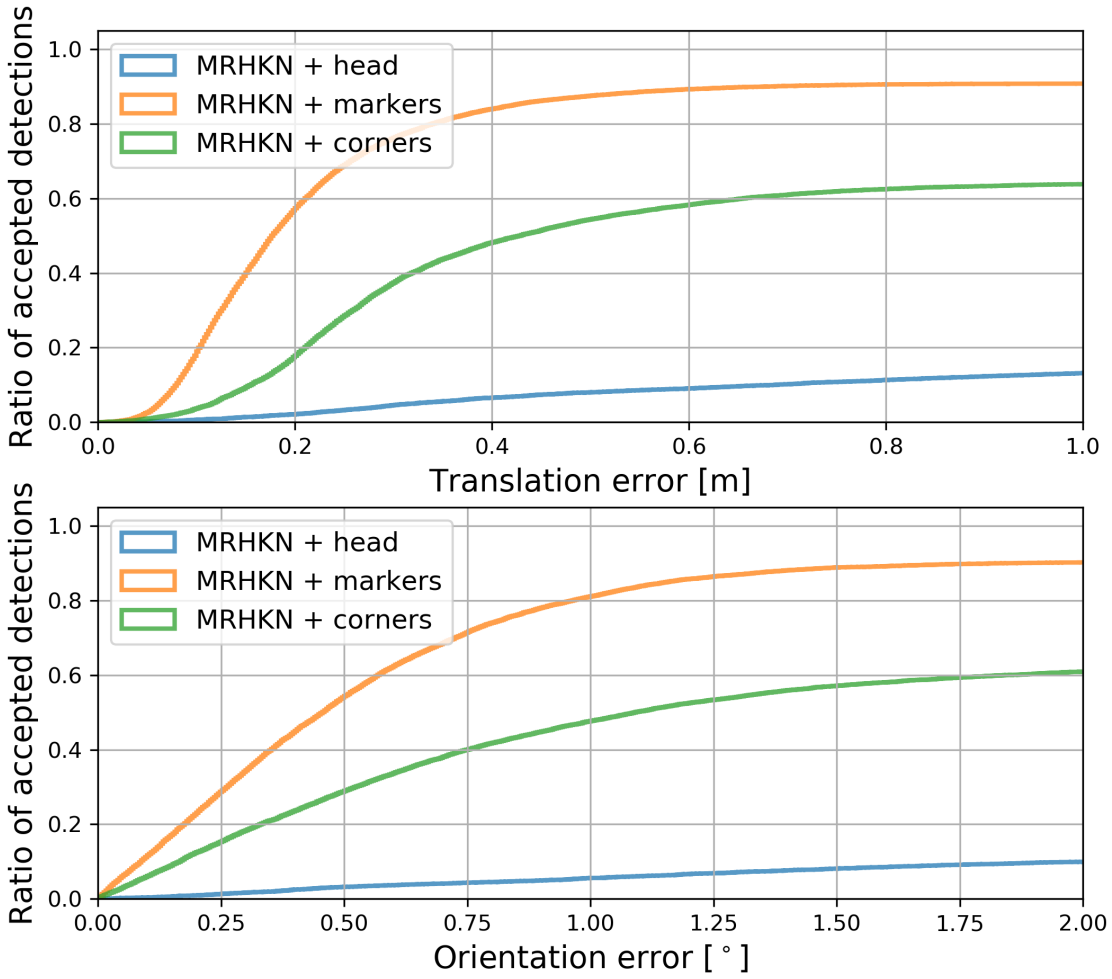

FIGURE 6.18: Cumulative distribution functions of 2D translation error (up) and orientation error (down) for the MRHKN approach with *head*, *markers*, and *corners* points for pose estimation. Version *head* performs poorly due to the small size of the object. Natural *corners* points work worse than the artificial *markers* approach.

6.4.5 Comparing the proposed models to existing solutions

The neural network architectures evaluated for comparison include HRNet and SCNet from the MMPose framework [67], the Faster R-CNN-based architecture from [78], and the ResNet101 backbone with the keypoint extraction head. These architectures were assessed uniformly, and the numerical results are summarised in Table 6.2. Overall, it was observed that results using natural *corners* were inferior to those obtained with artificial *markers*. Previous tests indicated poor performance for the *head* points arrangement with proposed network architectures, so this arrangement was excluded from the method comparison. When *markers* are used, the MRHKN approach shows the highest percentage of accepted detections, with the lowest median translation and rotation errors. Conversely, the RKN approach demonstrates the largest median translation and rotation errors among the evaluated solutions, while HRNet exhibits the lowest percentage of accepted detections.

The model from [78] could not operate with images sized 960×960 due to insufficient GPU RAM. This issue is intrinsic to the network architecture, as it generates heatmaps and two

offset maps (for the x and y axes) for each point, and then calculates the final location based on this data. Creating the required four-dimensional tensors, where each dimension matches the length of one side of the heatmap, demands substantial GPU memory. Consequently, the largest input image size manageable on a modern Nvidia A100 card with 40 GB of RAM was 500×500 pixels. In terms of processing time, the MRHKN network outperforms the Papandreou model while handling nearly double the image input size. When run on the Nvidia A100 GPU, Papandreou's network achieved 4 Frames Per Second (FPS), utilising 32 GB of RAM, whereas MRHKN achieved 5.5 FPS, requiring only 8 GB of GPU RAM.

TABLE 6.2: Comparison between the number of accepted detections, median 2D translation errors, and median 2D rotation errors of the RKNand MRHKN models and state-of-the-art HRNet, SCNet, ResNet101 (with head), and Papandreou's approaches (best results are bolded). Model marked with * was evaluated with input image size reduced to 500×500 due to memory limits.

Method	Version	Percent of accepted detections	Median t_{2D} [m]	Median r_{2D} [deg]
ResNet101	corners	39.4%	0.54	1.14
HRNet	corners	44.5%	0.43	1.52
SCNet	corners	18.0%	0.64	0.93
Papandreou*	corners	40.9%	0.53	1.15
RKN	corners	**84.5**%	0.92	1.96
MRHKN	corners	66.6%	**0.28**	**0.60**
ResNet101	markers	87.7%	0.30	0.75
HRNet	markers	82.0%	0.34	0.59
SCNet	markers	88.7%	0.31	0.59
Papandreou*	markers	63.6%	0.47	1.10
RKN	markers	88.7%	0.97	1.95
MRHKN	markers	**90.9**%	**0.17**	**0.41**

Figure 6.19 illustrates the performance comparison between RKN, MRHKN, the model from [78], and other state-of-the-art methods from the MMPose library, using images annotated with *markers* ground truth points. The *corners* arrangement was excluded due to its inferior performance compared to the *markers* arrangement. The histogram shapes indicate that the MRHKN method more accurately estimates both the position and orientation of the camera compared to the other methods. In contrast, the error distributions for the RKN and Papandreou's method are roughly linear and worse than those for the evaluated ResNet101, HRNet, SCNet, and the MRHKN models. The inferior performance of the state-of-the-art networks compared to the MRHKN solution is likely due to the different architecture of the keypoint head, which contains fewer convolution layers. The HRNet's lower performance might also be attributed to the relatively small number of feature maps returned by the backbone network. All MMPose-implemented methods (HRNet, SCNet, and ResNet101) perform similarly, with ResNet101 achieving the best translation error, while HRNet and SCNet exhibit smaller orientation errors.

Based on the evaluation presented, it is evident that the proposed MRHKN method surpasses state-of-the-art solutions and aligns best with the requirements. Consequently, MRHKN is the sole method considered for the docking scenario and is further assessed in the subsequent ablation study.

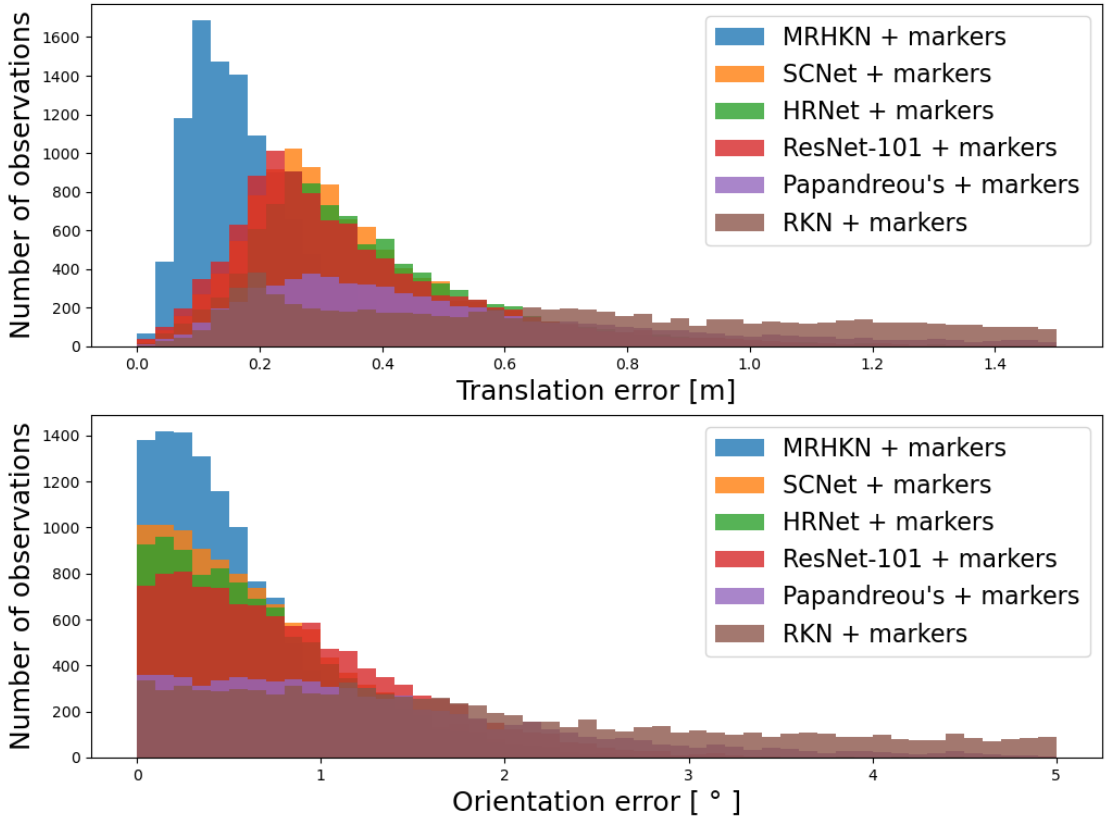

FIGURE 6.19: Histograms of 2D translation error (up) and orientation error (down) for the RKN and MRHKN approaches compared to the approaches implemented in the MMPose framework and Papandreou's network reveal superiority of the MRHKN approach. Model marked with * was evaluated with input image size reduced to 500×500 due to memory limits.

6.4.6 Influence of image resolution

The camera eventually installed in the buses by the operator may differ from the high-resolution camera used to acquire the training data for the previously discussed results. Consequently, during evaluation, the impact of image size on pose estimation accuracy was assessed (Fig. 6.20) to identify the minimum camera resolution that satisfies the localization accuracy requirements. To ensure consistency across different resolutions, a lower resolution camera was simulated by resizing the training and testing data to a fraction of the original size and adjusting the ground truth keypoint locations used for training accordingly.

Using this method, six different models were trained with progressively scaled-down image resolutions to evaluate performance with lower resolution images. The images were resized by scaling factors of 0.05 (*Scaled 0.05*), 0.1 (*Scaled 0.1*), 0.2 (*Scaled 0.2*), 0.4 (*Scaled 0.4*), 0.6 (*Scaled 0.6*), and 0.8 (*Scaled 0.8*). For each scaling factor, a new network was trained and tested on the appropriately resized images, with ground truth labels adjusted to match the discrete pixel values, accurately reflecting the training process on these scaled images.

The performance of the *MRHKN + markers* approach remains consistent with the original results in both accepted detection coverage and pose estimation error for the *Scaled 0.8*, *Scaled 0.6*, and *Scaled 0.4* versions. These versions demonstrate an accepted detection rate exceeding 90%, with

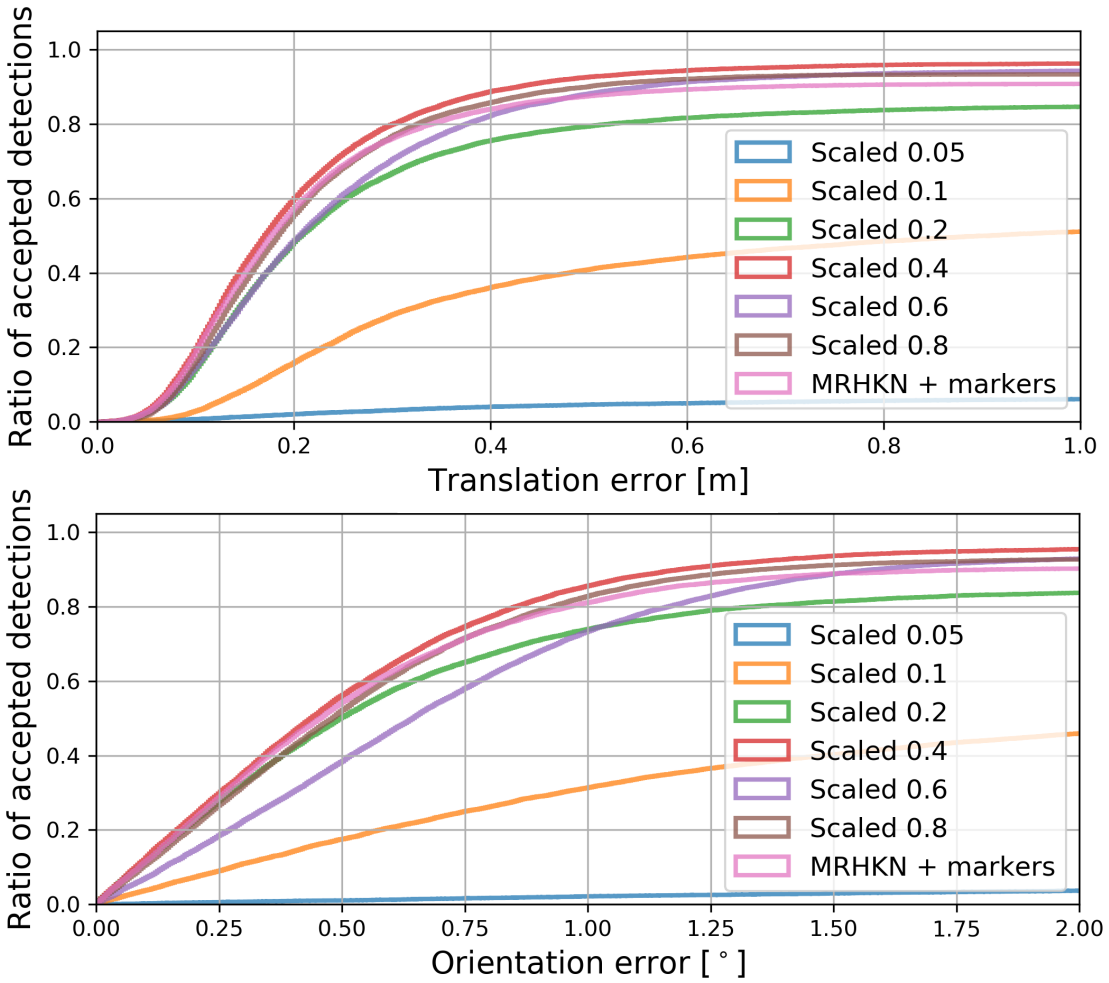

FIGURE 6.20: Cumulative distribution of translation and orientation errors for experiments with reduced image sizes for *MRHKN + markers* approach. No significant drop in performance even if the image is reduced up to 2188×1459 pixels was noticed.

median pose errors reported below 0.17 meters and 0.61 degrees. However, further reductions in image size result in noticeable performance declines. The *Scaled 0.2* version maintains median errors of 0.18 meters and 0.42 degrees, but the accepted detection rate falls to 85.5%. As anticipated, reducing the image size predominantly affects detection coverage and pose accuracy at greater distances. Additional reductions lead to poorer performance in the *Scaled 0.1* version and a significant breakdown for the *Scaled 0.05* version, which detects fewer than one-tenth of the charger keypoints identified by the original network with full-size images. Numerical results for all evaluated versions are summarised in Table 6.3.

Achieving satisfactory performance with reduced-resolution images allows for the implementation of a lower-resolution camera on the bus. This widens the selection of industrial-grade cameras suitable for the production system, taking into account factors such as interface compatibility and enclosure protection, while also lowering costs. Based on the experimental results, it can be concluded that a camera with a resolution of 2188×1459 (4 MP class) would be appropriate for the presented positioning system, providing accurate results at a fraction of the cost of the 20 MP camera used in the experiments.

TABLE 6.3: Performance comparison between versions of the *MRHKN + markers* method configured with different image sizes during training and testing

Scaling factor	Image size	Percent of accepted detections	Median t_{2D} [m]	Median r_{2D} [deg]
0.05	273×182	7.1%	0.35	1.95
0.1	547×364	56.4%	0.30	0.87
0.2	1094×729	85.5%	0.18	0.42
0.4	2188×1459	96.3%	0.17	0.42
0.6	3283×2188	94.8%	0.20	0.61
0.8	4377×2918	93.4%	0.18	0.45
1.0	5472×3648	90.9%	0.17	0.41

6.4.7 Performance dependence on the distance to the charging station

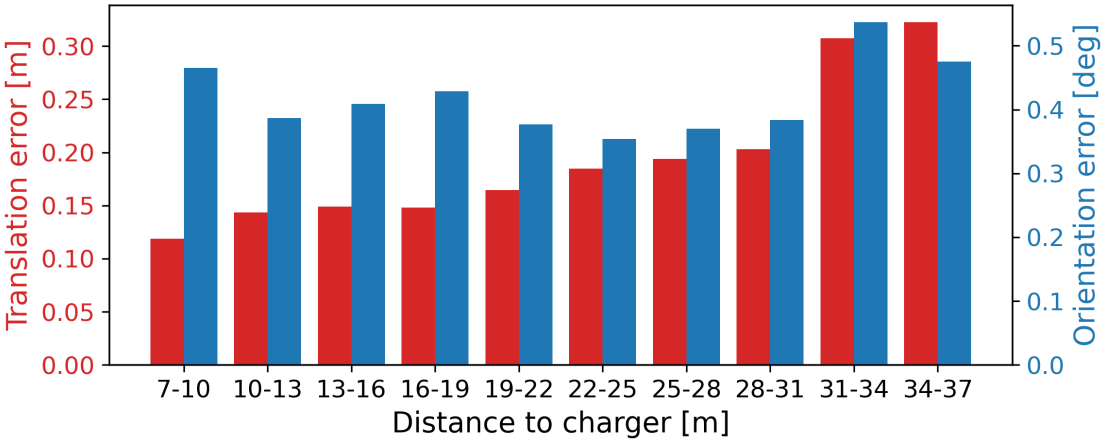

FIGURE 6.21: Distribution of the translation error (black bars) and orientation error (gray bars) as a function of the distance to the charging station for *MRHKN + markers*.

The distance to the observed object influences the accuracy of pose estimation. All accepted detections from the *MRHKN + markers* model were categorised into 10 bins, covering ranges from 7 to 37 meters, and the median errors for translation and rotation were calculated for each bin (Fig. 6.21). As expected, the translation estimation error increases with the observation distance, with a notable decrease in accuracy at distances greater than 30 meters. Despite the increasing translation error with distance, the rotation error remains relatively constant. These error characteristics are suitable for motion planning and execution procedures [65], as the rotation error remains low even beyond 30 meters, which is important for trajectory planning. The translation error decreases significantly as the bus approaches the charging station. Within the final few meters, the translation error reduces to about 10 cm, allowing the bus odometry to take over if the roof-mounted camera loses sight of all the markers. Once accurately positioned relative to the charger station head, and being very close, the bus can move along a straight path and safely engage the pantograph using its mechanical adaptation system.

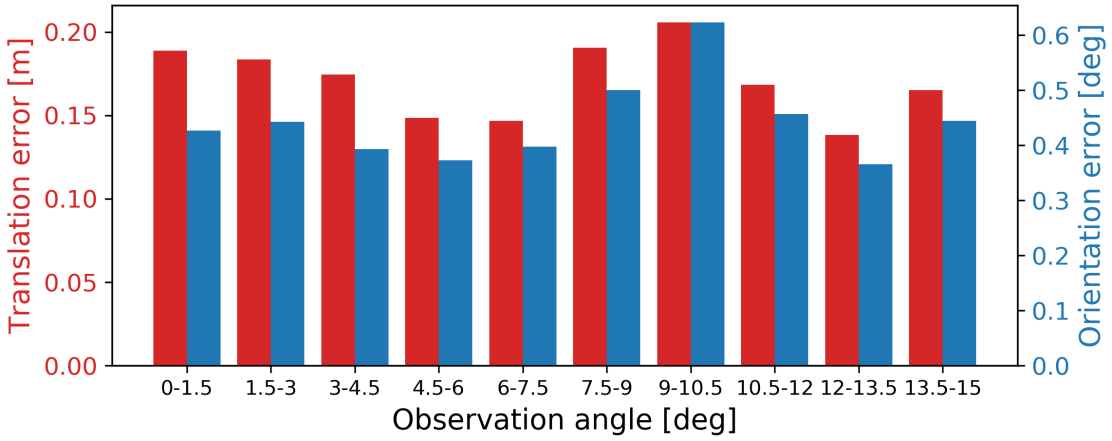

FIGURE 6.22: Distribution of translation error (black bars) and orientation error (gray bars) as a function of the orientation to the charger for *MRHKN + markers*.

6.4.8 Performance dependence on the observation angle of the charging station

Another factor that might influence the localization system's accuracy is the observation angle. In the test dataset, all observations were made at angles less than 15 degrees. This assumption is based on realistic maneuver scenarios where, at the beginning of the maneuver, the bus travels along a lane approximately parallel to the x-axis of the charger station's coordinate system. The observation angle largely depends on the lateral offset between the bus path and the roadside charging station (see Fig. 6.1). Similar to the previous analysis, all observations were divided into 10 bins, and the median error is shown in Fig. 6.22. The chart indicates that translation error is relatively unaffected by the observation angle, which is beneficial, as the system can handle less common scenarios with larger observation angles that might not be well-represented in the training dataset. Likewise, the accuracy of yaw angle estimation is not dependent on the observation angle. Comparing these error values to those in Fig. 6.21, it can be concluded that within the considered ranges, observation angle is a less significant factor in determining localization accuracy than distance.

6.4.9 Performance dependence on the bus speed

Docking the bus to a charging station is performed at a low speed due to the need for precise steering. The entire maneuver covers less than 40 meters and requires the bus to stop at the end. Bus operators typically instruct drivers not to exceed 20 km/h when approaching the charging station, but in practice, drivers tend to use much lower speeds for docking. The vision-based localization across the full range of speeds observed during the docking experiments was evaluated. As shown in Fig. 6.23, neither the translation nor the rotation estimation error is influenced by speed. Despite using a camera with a rolling shutter, the performance remained unaffected by vehicle motion, as the MRHKN approach ensures robust keypoint detection.

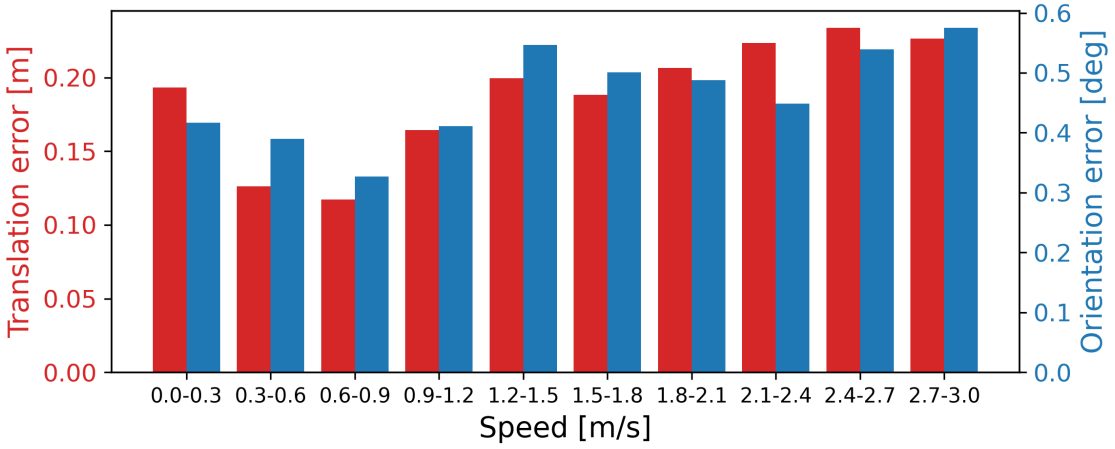

FIGURE 6.23: Distribution of translation error (black) and orientation error (gray) as a function of the bus speed for *MRHKN + markers*.

TABLE 6.4: Comparison of the size of networks, inference time, pose estimation errors, and percent of accepted detections of the evaluated architectures.

Heatmap size	Configuration	Operations [GFLOPs]	Parameters [M]	Inference time [ms]	Median t_2D [m]	Median r_2D [deg]	Percent of accepted detections
152×152	Papandreou	41.22	42.67	167.82	0.53	1.15	40.90 %
	Papandreou+Refined			171.82	1.16	1.67	52.34 %
128×128	Baseline	41.08	28.54	**35.60**	0.43	0.97	92.59 %
	Refined			40.60	0.44	0.98	95.66 %
	GAKN			**35.60**	0.37	0.67	92.14 %
	GAKN+Refined			40.60	0.37	0.67	94.79 %
256×256	Baseline	43.34	28.67	38.40	0.35	0.74	93.56 %
	Refined			42.40	0.36	0.74	95.99 %
	GAKN			38.40	0.32	0.62	95.56 %
	GAKN+Refined			42.40	0.32	**0.61**	**96.60** %
512×512	Baseline	112.47	29.72	50.60	0.32	0.70	94.03 %
	Refined			54.60	0.32	0.71	95.02 %
	GAKN			50.60	**0.30**	0.64	92.82 %
	GAKN+Refined			54.60	0.31	0.64	94.44 %

6.4.10 Geometry-Aware Keypoint Network

In this section, the results of the evaluation of the Geometry-Aware Keypoint Network will be presented. For all experiments, the input image size was set to 512×512 pixels. The results of the main experiments are shown in Table 6.4. The configurations of the keypoint extraction network considered include the Baseline (HRNet32 with a 3-layer head), Refined (Baseline with Reprojection-based Pose Refinement), GAKN, and GAKN+Refined (GAKN with Reprojection-based Pose Refinement).

Backbone network and Keypoint Head depth

Comparing the HRNet32 and HRNet48 networks (Table 6.5), it is evident that the medians of rotation and translation errors are lower for the HRNet48 network across both keypoint head variants. However, there is no significant impact on the percentage of accepted detections. The inference time is slightly longer for the HRNet48 version, and the required operations and number of parameters are more than double those of the HRNet32 backbone. Additionally, the

TABLE 6.5: Comparison of the size of networks, inference time, pose estimation errors, and percent of accepted detections of the networks with different backbones and depths of the keypoint head.

Heatmap size	Backbone	Convolution layers in the keypoint head	Operations [GFLOPs]	Parameters [M]	Inference time [ms]	Median t_2D [m]	Median r_2D [deg]	Percent of accepted detections
128×128	HRNet32	1	41.08	28.54 M	**35.40**	0.46	0.98	93.59 %
	HRNet48		84.10	63.60 M	37.10	0.42	0.72	92.83 %
	HRNet32	3	41.12	28.54 M	35.60	0.43	0.97	**93.89 %**
	HRNet48		84.18	63.60 M	37.60	**0.40**	**0.71**	92.87 %

HRNet48-based network demands over 8 GB of GPU memory for the entire processing pipeline. For commercial applications, the cost of the hardware must be considered. Despite the improved pose estimation accuracy with HRNet48, the HRNet32 backbone was chosen for the Baseline due to its acceptable accuracy and compatibility with low-end hardware (GPU). Consequently, the GAKN architecture configurations evaluated in the experiments also utilise the HRNet32 backbone.

Adding extra convolutional layers to the keypoint head slightly reduces the median translation and rotation errors (Tab. 6.5). This modification does not impact the percentage of accepted detections and has only a minimal effect on the number of operations and inference time.

Heatmap size

In this subsection, the performance of the Baseline network across three different heatmap resolutions will be compared. The default keypoint detector implementation based on HRNet generates heatmaps downsampled by a factor of four relative to the input image size, resulting in 128×128 pixel heatmaps for a 512×512 pixel image. Increasing the heatmap size has a negligible effect on the percentage of accepted detections but significantly reduces translation and rotation errors. The increase in the number of operations is modest when moving from a 128×128 to a 256×256 heatmap; however, for the 512×512 version, the number of operations triples. Consequently, the processing time for a single image doubles in the 512×512 version compared to the 128×128 version. While larger heatmaps improve location accuracy, they also increase processing time. Nonetheless, the increase in network parameters with larger heatmaps is minimal.

This indicates that all the discussed architectures, even the largest one, can fit on a graphics card with 8 GB of memory. As the heatmap size increased, a reduction in errors was observed, specifically, the translation error of a 128-sized heatmap compared to a 512-sized heatmap was reduced by 26.7% and the rotation error by 33.9%. For the 256-sized heatmap, errors were higher than the 512-sized but lower than the 128-sized heatmap, confirming the relationship between heatmap size and location accuracy. Heatmap size did not affect the percentage of accepted detections.

The GAKN network configuration maintains the same number of operations, parameters, and processing times because modifications were only made during training and do not affect inference. There is a reduction in translation and rotation errors across all three heatmap sizes. When using reprojection loss with a 128-sized heatmap, there was a 15.2% reduction in translation error and a 37% reduction in rotation error. For the 256-sized heatmap, translation error

decreased by 8.4% and rotation error by 15.5%. For the 512-sized heatmap, there was a 2.6% reduction in translation error and a 9.5% reduction in rotation error.

Combining both approaches: increasing heatmap size and applying reprojection loss, resulted in the lowest median translation error among all tested models and a median rotation error comparable to the best obtained with the 256-sized heatmap. Using a 256-sized heatmap offers a reasonable trade-off between processing time and location accuracy. The system achieved the best rotation error result, while the translation error was comparable to the best result, and exhibited a short inference time similar to the 128×128 heatmap. Additionally, the percentage of accepted detections in the GAKN configuration with this heatmap size is higher than in the baseline.

6.5 Pose estimation of surrounding cars for autonomous driving

In this section, the evaluation of the pipeline for estimation of the pose of surrounding cars will be presented.

Model evaluation was performed on the validation set of the ApoloCar3D dataset, which includes 200 images as defined by the dataset's authors. On a Nvidia 1080Ti GPU, the proposed pipeline achieves a processing speed of 20 FPS for the EPNP variant without uncertainty propagation, and 18 FPS when full uncertainty propagation is included.

6.5.1 Keypoints 2D

For the evaluation of the 2D keypoints estimation network, the PCK metric was used with three thresholds: 5 pixels, 10 pixels, and 15 pixels.

The first two rows in Table 6.6 were computed using only visible vehicle points as references, specifically those included in the ground truth annotations of the dataset. The metric values for all marked points were presented as well as for those points that the network identified as having the highest Keypoint Score Head score. The results demonstrated that the network accurately assesses the point estimation accuracy and is capable of improving the PCK metric results.

The last two rows in Table 6.6 show similar PCK metric values but with a key difference: the ground truth annotations used were the projections of 3D points, based on the ground truth translation and rotation of each vehicle. This method allows for the evaluation of invisible points, which is more relevant to real-world scenarios where the visibility of points is not known in advance.

The right part of the table presents results obtained using a modified network architecture, where the HrNet backbone network was replaced by the ViTPose model.

The results, particularly for the points selected by the KSH score, indicate sufficient accuracy for 3D pose estimation using PnP algorithms.

TABLE 6.6: PCK metric of the 2D keypoints accuracy comparing to manually labelled visible points and all 66 points acquired by the projection of 3D points, for HRNet estimates (top) and ViTPose estimates (bottom).

Threshold	5 px	10 px	15 px	5 px	10 px	15 px
	HrNet			ViTPose		
Visible all	44.9	71.0	81.7	67.2	85.1	90.8
Visible selected	64.0	87.2	93.6	74.5	**90.8**	**94.8**
All	19.0	32.7	41.0	28.3	39.4	45.0
Selected all	59.6	82.6	89.9	**76.6**	89.4	93.5

6.5.2 Keypoints 3D

In this section, the results related to estimating 3D points on vehicles using the proposed method will be presented. The module's performance is assessed using the MPJPE metric, which quantifies the average distance error of the estimated points. The findings show an average MPJPE value of 0.119 m for all points estimated by the network. Focusing on the N points with the highest KSH scores reduces the error to 0.105 m, confirming that the KSH score improves keypoint quality. The most accurately estimated points are located on the rear corner of the car handle of the right front door, with a mean error of 0.073 m, while the least accurate points, averaging an error of 0.235 m, are found on the left corner of the rear bumper.

6.5.3 A3DP metrics

The Absolute Average 3D Precision (A3DP-Abs) metric, as introduced in [93], was employed to evaluate the results. This metric emphasizes the absolute distances to objects and considers three components: the estimated shape, position, and rotation of the car. The translation error metric is defined as:

$$c_{\text{trans}} = \left|\mathbf{t}_{gt} - \hat{\mathbf{t}}\right|_2 \leq \delta_t, \tag{6.2}$$

where $\mathbf{t}_{gt}$ denotes ground truth translation, $\hat{\mathbf{t}}$ denotes estimated translation and δ_t is an acceptance threshold. The rotation error metric is defined as:

$$c_{\text{rot}} = \arccos\left(\left|\mathbf{q}_{gt} \cdot \hat{\mathbf{q}}\right|\right) \leq \delta_{rot}, \tag{6.3}$$

where $\mathbf{q}_{gt}$ denotes ground truth rotation quaternion, $\hat{\mathbf{q}}$ denotes estimated rotation quaternion, and δ_{rot} is an acceptance threshold.

Similar to the metrics proposed for the COCO dataset, the authors of ApolloCar3D introduced a set of metric thresholds ranging from strict to loose. The translation thresholds range from 2.8 m to 0.1 m, increasing by 0.3 m increments, while the rotation thresholds range from $\pi/6$ to $\pi/60$ in steps of $\pi/60$. Besides the 'mean' metric, which averages results across all thresholds, two single-threshold metrics were defined. The loose criterion (denoted as $c-l$) uses [2.8, $\pi/6$] thresholds for translation and rotation, whereas the strict criterion (denoted as $c-s$) employs [1.4, $\pi/12$] thresholds. For evaluating the 3D shape reconstruction, a predicted mesh is rendered from 100 different perspectives, and the Intersection over Union (IoU) is computed between these

renderings and the ground truth masks. The average of these IoU values is then used to assess the shape reconstruction accuracy.

Table 6.7 presents the results and compares them with state-of-the-art methods. For a fair comparison, the implementations of the algorithms proposed in [50] and [14] were taken from the baselines provided in [93].

TABLE 6.7: Comparison of results with the state-of-the-art methods on A3DP-Abs metrics

algorithm	mean	$c-l$	$c-s$
3D-RCNN [50]	16.4	29.7	19.8
DeepMANTA [14]	20.1	30.7	23.8
GSNet [38]	18.9	37.4	18.4
BAAM-Res2Net [52]	25.2	47.3	23.1
Ours EPnP	23.4	44.6	31.7
Ours BFGS	**25.6**	**47.7**	**34.6**

The results of the A3DP-Abs mean metric demonstrate that the implemented system outperforms the recently proposed BAAM method [52], showcasing the proficiency of the selected solution in accurately estimating the 3D position of surrounding vehicles. A significant advantage of the network is its performance on the strict criterion $c-s$, which evaluates the module's ability to estimate 3D points under stringent conditions. This challenging task requires a high level of precision and reliability. This system surpasses all existing state-of-the-art solutions on this metric by a considerable margin, underscoring its excellence in providing highly accurate 3D characteristic points.

The table 6.8 presents the results of an ablation study that examines the influence of a backbone network, the selection of points for the PnP algorithm, and the accuracy of keypoints 3D estimation.

The first set of variants uses a transformer-based network, called ViTPose, as the backbone for feature extraction. The study investigated the influence of point selection on the accuracy of solutions to the PnP problem. One variant used 17 points known to have the lowest estimation error relative to the ground truth labels. The other set of variants relied on points identified as visible by annotators, representing a more intuitive method of point selection that emphasises the reliability of point visibility in pose estimation. The purpose of these comparative studies was to assess how the accuracy of 2D point estimation and point selection affect the performance of algorithms solving the PnP problem. Through these comparisons, the goal is to discover the most effective strategies for improving the accuracy of vehicle pose estimation using deep neural networks.

The network using HRNet as the backbone, together with the 17 best-estimated points (HRNet Best 17), slightly outperformed the variant using the ViTPose network (ViTPose Best 17) and achieved results very similar to the baseline variant, where the points for solving the PnP problem are selected by a neural network. This indicates that the quality of point estimation by the neural network is close to optimal for solving the PnP problem. Despite the superior accuracy of the ViTPose network in 2D point estimation, its performance in pose estimation was not as strong. This discrepancy could be due to the accuracy of the corresponding 3D point estimates and their spatial distribution on the vehicle.

TABLE 6.8: Comparison of results from different variants on A3DP-Abs metrics. $c-l$ is a loose criterion that uses thresholds=[2.8 m, $\pi/6$ rad], while $c-s$ is a strict criterion that uses thresholds=[1.4 m, $\pi/12$ rad] for translation and rotation errors respectively.

Algorithm	mean ↑	c-l ↑	c-s ↑
Baseline	25.6	47.7	34.6
ViTPose Best 17	25.7	46.7	34.7
HRNet Best 17	27.6	49.8	38.4
ViTPose Best 17 GT J3D	**46.9**	68.5	**60.5**
HRNet Best 17 GT J3D	46.7	**68.7**	60.4
ViTPose Visible	22.4	41.3	30.6
HRNet Visible	18.7	35.8	26.4
VitPose Visible GT J3D	33.1	51.9	44.1
HRNet Visible GT J3D	28.5	48.8	39.3

Further investigation included a pair of variants, named ViTPose Best 17 GT J3D and HRNet Best 17 GT J3D for pose estimation, which used ground truth coordinates for 3D points and the 17 best estimated 2D points. In this case, the ViTPose network performed slightly better, supporting the idea that the most accurately estimated 2D points by HRNet are also linked to the most accurately estimated 3D points. This success is likely influenced by the use of reprojection loss during training, highlighting the critical role of reprojection loss in training for the accurate estimation of 3D points from their 2D counterparts.

Another variant analysis used 3D points estimated by the network and selected points marked as visible for pose estimation to remove the influence of point selection on estimation accuracy. Here, the ViTPose network performed better than HRNet, which showed a significant decrease in accuracy with HRNet Visible compared to HRNet Best 17. In contrast, the ViTPose network showed a much smaller decrease, highlighting the impact of joint learning for the branches estimating 2D and 3D points, as opposed to the ViTPose variant where the 2D point estimation network was trained independently. This demonstrates the importance of integrated training approaches and suggests that they may be more effective in maintaining accuracy when factors such as point visibility are taken into account.

The final variant analysis included ground truth 3D points and points marked as visible. The ViTPose network again showed superior performance, although the results were not as robust as the ViTPose Best 17 GT J3D and HRNet Best 17 GT J3D variants. This observation highlights the importance of the spatial distribution of the selected points and the need to include points that are not marked as visible for pose estimation, highlighting how important point selection and spatial distribution are to the accuracy of pose estimation tasks.

6.6 Car shape estimation

The table 6.9 compares an approach described in Section 4.3 that directly estimates the positions of 3D keypoints to an approach that first estimates a dense mesh described in 4.6, from which keypoint coordinates are then regressed. The first column compares the MPJPE for all 66 keypoints on the ApolloCar 3D validation dataset. The results show that the mesh-based approach

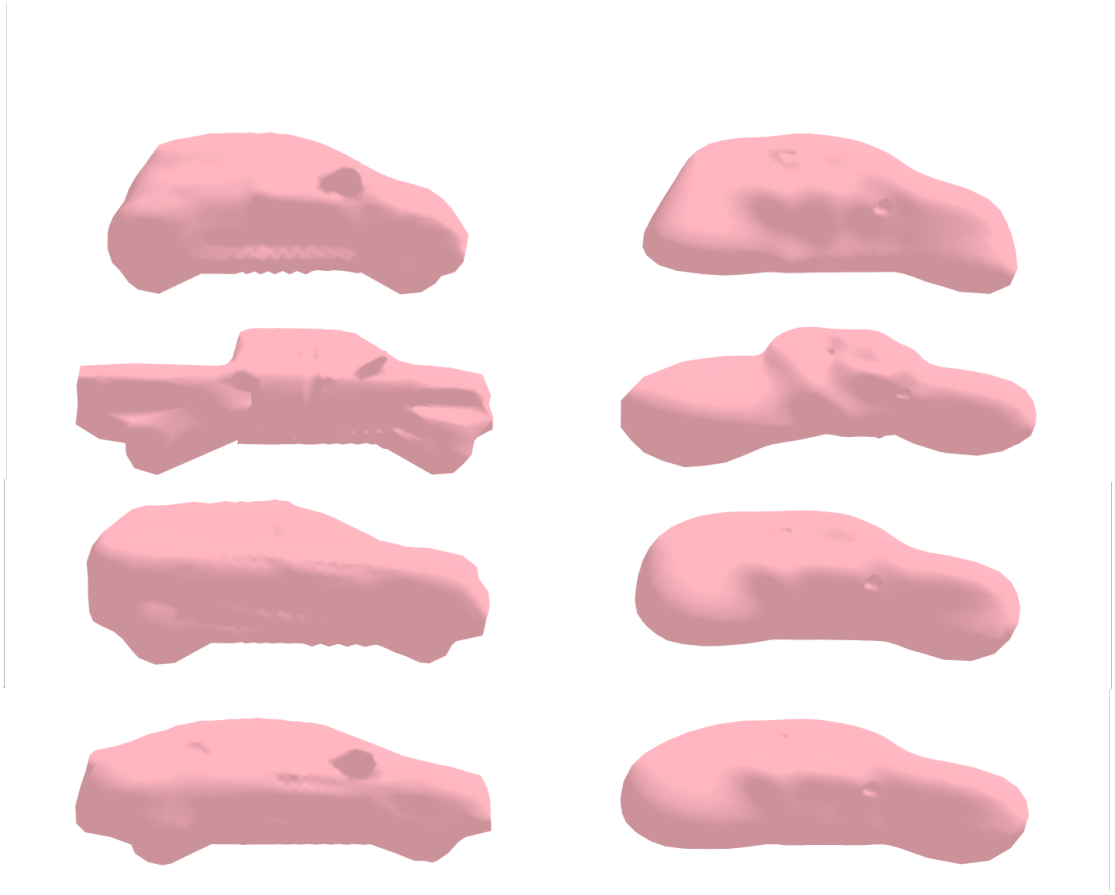

FIGURE 6.24: Ground truth meshes (left column) and estimated counterparts (right column)

performs slightly better. However, when comparing the A3DP-Abs metrics, the mesh-based approach yields worse results. An explanation can be found by examining the MPJPE for the 17 keypoints selected by the Keypoint Score Head. Here, it is evident that the accuracy in the case of the mesh-based approach did not improve, unlike the direct approach, which exhibits smaller estimation errors. The likely cause of these results is that the accuracy of keypoint estimation in the mesh-based approach does not depend on the viewpoint, because their coordinates are extracted directly from the mesh rather than the image. This means that all points have similar estimation errors. On the other hand, the direct approach can more precisely estimate points that are actually visible, and the variation in estimation errors is larger, thus making it possible to find a subset of 17 points that has a lower mean estimation error compared to the mesh-based approach. On the Fig. 6.24 are shown examples of car shape reconstructions.

TABLE 6.9: Comparison of Keypoint 3D MPJPE and A3DP-Abs metric for Keypoint 3D based and Dense mesh based pipelines for vehicles pose estimation.

	Keypoint 3D MPJPE All	Keypoint 3D MPJPE Top 17	A3DP-Abs mean	A3DP-Abs c-l	A3DP-Abs c-s
3D keypoint based	0.119	0.105	25.6	47.7	34.6
Dense mesh based	0.111	0.110	23.3	43.7	32.6

TABLE 6.10: Percentage of ground-truths within uncertainty ellipse for different standard deviations.

Standard deviation	1σ	2σ	3σ
Percentage of ground-truths within uncertainty ellipse	48.75 %	87.38 %	98.00 %

6.7 Uncertainty estimation

This section presents an experimental evaluation of the uncertainty estimation methods introduced in the previous chapters of the dissertation. The focus is placed on two distinct scenarios: keypoint estimation uncertainty in the context of docking to the charging station and uncertainty of pose estimation of surrounding vehicles.

The quality of the estimated uncertainty for 2D points detected on the charger was evaluated using a hand-labeled validation dataset of approximately 200 images. The geometric uncertainty prediction was assessed by measuring the percentage of ground truth keypoint locations that fall within the 1, 2, and 3 σ uncertainty ellipses. The numerical values for these evaluations are presented in Tab. 6.10.

Qualitative results of the covariance matrices prediction are illustrated in Fig. 6.25. Comparing Fig. 6.25A and Fig. 6.25B reveals that keypoint detection uncertainty decreases as the distance to the charger decreases. Figures 6.25C and 6.25E demonstrate increased uncertainty for poor quality images. In Fig. 6.25D, the two leftmost points exhibit greater uncertainty along the x-axis because, from that viewpoint, estimating their precise location is more ambiguous.

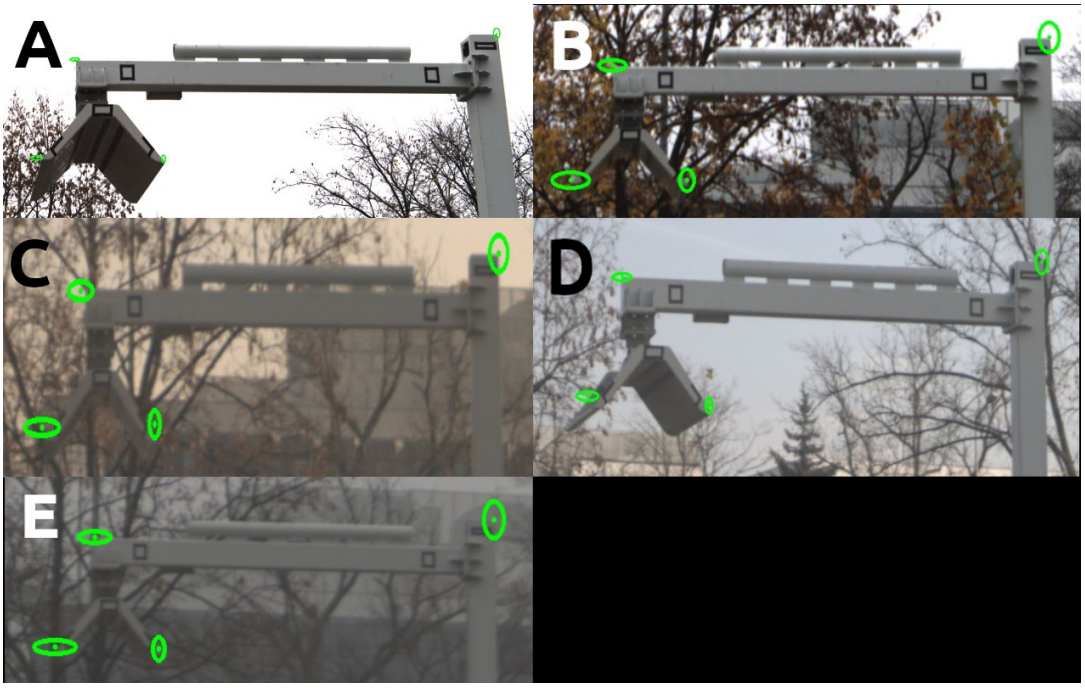

FIGURE 6.25: Visualisation of estimated keypoints locations and 3 sigma uncertainty ellipses for different observation cases. Fig. A presents detection from a close distance, B - a far distance, C - a blurry image, D - a different observation angle, and E - cloudy/foggy weather.

The following fragment presents results related to the uncertainty estimation of 2D and 3D characteristic points, as well as the vehicle's pose using the ApolloCar3D dataset. The visualizations of 2d keypoints uncertainty are shown in Fig. 6.26.

FIGURE 6.26: Visualization of 2D keypoints uncertainty represented by ellipses, each corresponding to a 1-σ standard deviation from the estimated values.

The initial evaluation analysed the percentage of point and vehicle translation estimations falling within the ranges of 1, 2, and 3 standard deviations (σ). These results are detailed in Table 6.11. Additionally, the relationship between the mean value of σ and the vehicle's distance is explored, as illustrated in Fig. 6.27.

These findings indicate that as the distance increases, the uncertainty in pose estimation also rises. This trend is expected, as distant objects appear with lower resolution in images, leading to higher uncertainties in estimation. Another notable observation is that the uncertainty along

TABLE 6.11: Percentage of estimation that falls within the ranges of one, two, and three σ

	Keypoints 2D		Keypoints 3D			Pose translation		
	x	y	x	y	z	x	y	z
1 σ	79.0	79.8	82.5	80.3	64.8	84.5	85.0	81.3
2 σ	93.1	93.8	94.2	94.1	80.9	94.9	95.6	93.7
3 σ	97.2	97.2	97.5	97.3	87.5	98.6	98.6	97.5

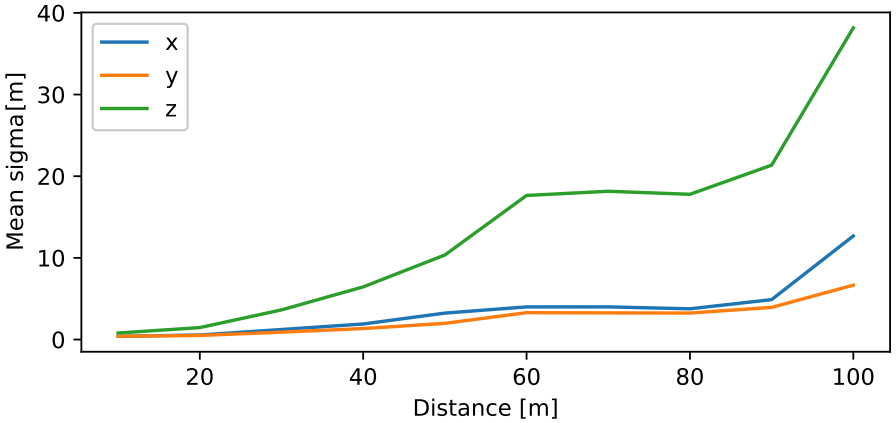

FIGURE 6.27: Plot of the mean σ (standard deviation) depending on the observation distance on the validation subset of the ApolloCar3D dataset.

the z-axis (depth direction) is greater than along the other two axes. This is understandable, as determining the position along the axis perpendicular to the image plane is often more challenging. Due to the nature of monocular vision and image projection, depth information tends to be less reliable, resulting in higher uncertainties. Figures 6.28 illustrates six examples of predictions and the associated uncertainty of the estimated pose from a bird's eye view. The uncertainty in position (x, y) for each vehicle is depicted by its uncertainty ellipse. It is evident that cars that are partially occluded or located further from the camera have larger uncertainty ellipses compared to those that are closer and fully visible.

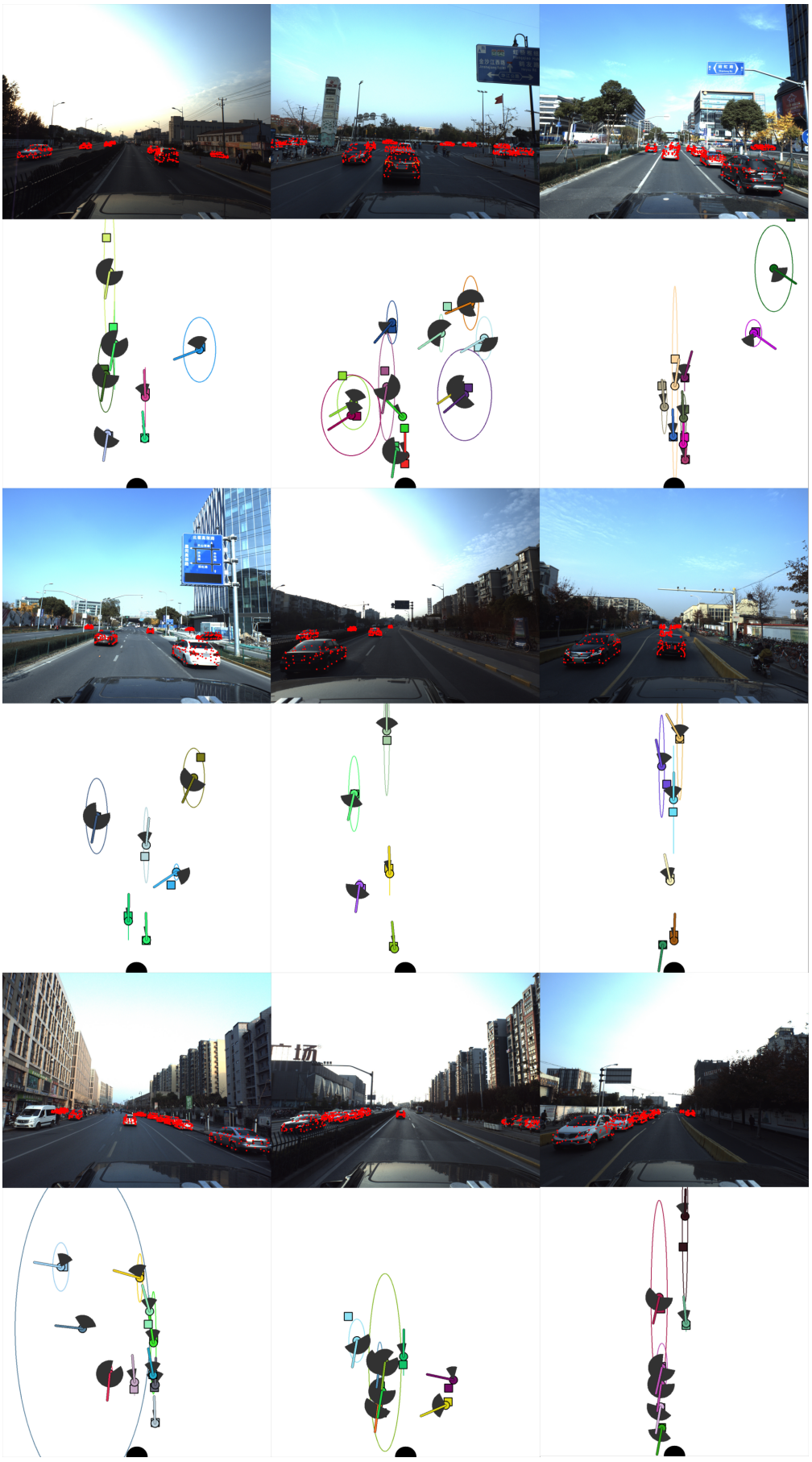

FIGURE 6.28: Visualizations of the estimated pose and uncertainty are presented. Circles indicate the estimated positions of the cars, while squares represent their ground truth positions. Colored rays depict the ground truth yaw angles, and grey fans illustrate the estimated one-sigma yaw range. Ellipses denote the one-sigma position uncertainty.

Chapter 7

Conclusions

7.1 Summary

This dissertation has addressed challenges in the field of data-efficient and explainable machine learning within the context of visual perception for autonomous vehicles. The research set out with specific goals to enhance the functionality and efficiency of deep neural networks by focusing on four primary objectives: efficient training on limited datasets, enhancing the explainability and interpretability of Deep Neural Networks outputs, designing a DNN architecture for estimating 3D geometric features from monocular images, and improving DNN performance by incorporating geometric constraints.

Chapter 2 introduced contributions to the field of object detection in autonomous driving. Initially, it presented **a novel method for the visualization of spatial areas of interest within the Faster R-CNN network**. Additionally, the chapter detailed **a guided learning procedure that was designed to enhance the performance of machine learning models when training data was scarce**. This approach utilised targeted feedback from the visualization method to identify shortcomings in the model's training regime and augmented the training dataset with appropriate examples.

Chapter 3 highlighted advancements in neural network design and processing techniques within the context of the estimation of semantic keypoints location. **Three new neural network architectures were introduced, specifically tailored for the precise detection of keypoints on an electric bus charger**. The architectures were designed to preserve high spatial resolution throughout the processing pipeline, which is needed for maintaining the precision of keypoint localization. In addition to the architectural innovation, **a new loss function was developed that leverages knowledge of the 3D object model and geometrical environment constraints**. By incorporating 3D object models directly into the training process, the network learned to predict keypoints that are geometrically consistent with the real-world structure of the objects, thereby reducing estimation errors. Furthermore, this chapter presents **a neural network branch designed to estimate the uncertainty of keypoint coordinate**

predictions. By integrating uncertainty estimation directly into the neural network architecture, the system provides a probabilistic measure of confidence in each prediction. This feature is particularly valuable in autonomous driving applications, where decision-making processes must consider the reliability of sensory and perception data to ensure safe and efficient operation.

Chapter 4 introduced contributions in neural network architectures and automated labelling techniques, aimed at enhancing the precision of 3D spatial estimations from single images. Firstly, **a new neural network architecture was developed for the estimation of characteristic points of cars in 3D space** from a single image. This network allowed for the estimation of 66 vehicles' semantic keypoints with a mean error of less than 12 cm. Additionally, an **automatic procedure for obtaining labels of car's characteristic points in 3D space** was introduced. This procedure automated the typically labor-intensive and error-prone task of manually labelling 3D points, thereby streamlining the training process and enhancing the dataset's quality. By automating this process, the network could learn from a consistently accurate and richly annotated dataset, leading to improvements in model performance and reliability. Furthermore, the chapter detailed the development of **a new neural network architecture designed specifically for estimating a dense 3D mesh of cars** from a single image. This architecture allowed for a reconstruction of vehicle geometry, which can be applied to applications requiring high fidelity spatial information such as advanced driver-assistance systems and fully autonomous driving solutions.

Chapter 5 presented **a processing pipeline designed for the estimation of the pose of surrounding cars using a single monocular image**. The pipeline integrates image processing and machine learning techniques to determine the position and orientation of nearby vehicles, a feature required for ensuring safe navigation and interaction in dynamic traffic environments. Additionally, the chapter introduced **a pose refinement procedure for correcting pose estimation errors** that often arise from outlier characteristic point estimations. This refinement procedure effectively adjusts the estimated pose by identifying and mitigating the impact of anomalous data points, thereby enhancing the reliability and accuracy of the pose estimation process. Furthermore, the chapter detailed **the application of the unscented transform to propagate the uncertainty of keypoint estimations to the overall uncertainty of the estimated pose**. This mathematical approach allows to estimate pose uncertainty from the keypoint 2D and 3D uncertainty by considering the non-linearities in the transformation from image space to pose estimation. By implementing the unscented transform, the pipeline provides a quantifiable measure of confidence in the pose estimations generated by the system, offering a more robust and reliable system for autonomous driving applications.

Chapter 6 detailed the **practical verification of attention visualization and guided learning procedure**. Visualisation of the network's attention allowed for deeper insights into model behavior, which is useful for debugging and refining model performance. The application of guided learning reduced the false positive detections without the need to acquire more data from the target operational environment. This is particularly beneficial for niche applications in autonomous driving such as docking to the charging station with electric city buses where acquiring large labeled datasets can be expensive and time-consuming. This chapter presented also the **practical verification of pose estimation methods based on deep neural networks**

that incorporate uncertainty measurement. These methods were tested on real-world applications, including the docking of electric buses to charging stations and the estimation of the pose of surrounding cars using a realistic dataset. The methods were successfully applied to detect and estimate the pose of a bus in a bus charger coordinate system from a distance greater than 30 meters, using a monocular camera. Based on these estimations, it was possible to plan and execute a docking maneuver with an error margin of less than 30 cm. Additionally, the pose estimation methods were applied to the task of determining the pose of surrounding vehicles in real-world driving conditions. The proposed method achieved state-of-the-art results on the public dataset ApolloCar3D, showcasing their robustness and accuracy.

7.2 Conclusions and thesis contribution

The algorithms presented in this dissertation evaluated experimentally on practical tasks make contributions to the current state of the art in machine learning for autonomous driving. The following contributions support the research theses:

1. A method for the visualization of spatial areas of interest within the Faster R-CNN object detection network coupled with a guided learning procedure designed to enhance the performance of machine learning models when training data was scarce.
2. A neural network that estimates uncertainty of keypoints 2D and 3D and method for propagating these estimations to the pose uncertainty.
3. Three new neural network architectures tailored specifically for the precise detection of keypoints on an electric bus charger.
4. A neural network capable of estimating the 3D car keypoint coordinates and shape represented as a dense mesh
5. A method for automatic labelling of keypoints 3D.
6. A loss function that leverages geometric constraints to improve the keypoint detection network's output.
7. A keypoint postprocessing procedure that corrects imprecise keypoint detections based on geometric constraints.
8. A processing pipeline designed for the estimation of the pose of surrounding cars using a single monocular image with pose uncertainty estimation

Considering the contributions presented above and the results of the experimental evaluation of the aforementioned algorithms, the theses stated at the beginning of the dissertation can be verified:

1. The results of experimental evaluations presented in Section 6.3 support the first hypothesis, i.e. "Deep learning architectures that allow us to extract and visualise meaningful

intermediate features make it possible to guide learning by augmenting the existing data sets."

2. The results of experimental evaluations presented in Section 6.7 supports the second hypothesis i.e. "Deep learning architectures allow us to describe the uncertainty of the geometric features produced by the network."

3. The results of experimental evaluations presented in Sections 6.5.2 and 6.6, and the description of an algorithm in section 4.4 that led to results presented in section 6.5.3 support the third hypothesis, i.e. "Using the available 3D models of observed objects makes it possible to learn geometric features from 2D images of these objects without exact labelling, and improves geometric feature detection from 2D images of these objects."

4. The results of experimental evaluations presented in Sections 6.4.10 and 6.5 support the fourth hypothesis, i.e. "The knowledge of the geometric constraints stemming from known object models allows a deep learning architecture to decrease the number of falsely detected features and increases the accuracy of feature location."

7.3 Future work

Building on the contributions of this dissertation, there are several avenues for advancing the research in data-efficient and explainable machine learning for visual perception within autonomous vehicles.

There is an opportunity to enhance the resolution and detail of car shape reconstructions by integrating generative neural network models, such as diffusion models, into the current network framework. By leveraging the capabilities of these novel generative models, it may be possible to produce more detailed and accurate representations of vehicle geometries, which could improve the overall effectiveness of the shape estimation process.

Enhancing the generalization capabilities of the pose estimation pipeline for surrounding vehicles presents another promising area for future work. Utilising Variational Autoencoders for car shape compression offers a novel approach to generate diverse vehicle shapes without requiring extensive dataset expansions. By sampling and decoding from the latent space of a VAE, diverse car shapes can be synthesised, potentially enhancing the robustness and adaptability of the pose estimation models to different vehicle types and configurations.

Future research could also explore the integration of more complex geometric constraints into the pose estimation process. This could include considering the relative pose of other vehicles in the vicinity and incorporating temporal context to enhance the accuracy of pose estimations. By understanding the interactions and relative positions of multiple vehicles, the pose estimation algorithms could achieve higher precision and reliability, particularly in complex driving scenarios.

Bibliography

[1] ZED-F9P, u-blox F9 high precision GNSS module, 2020. URL https://www.u-blox.com/sites/default/files/ZED-F9P_DataSheet_%28UBX-17051259%29.pdf.

[2] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[3] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2D human pose estimation: New benchmark and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3686–3693, 2014.

[4] Plamen P Angelov, Eduardo A Soares, Richard Jiang, Nicholas I Arnold, and Peter M Atkinson. Explainable artificial intelligence: an analytical review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 11(5):e1424, 2021.

[5] Simone Antonelli, Danilo Avola, Luigi Cinque, Donato Crisostomi, Gian Luca Foresti, Fabio Galasso, Marco Raoul Marini, Alessio Mecca, and Daniele Pannone. Few-shot object detection: A survey. *ACM COMPUTING SURVEYS*, 54(11S), JAN 2022. ISSN 0360-0300. doi: 10.1145/3519022.

[6] Eduardo Arnold, Omar Y Al-Jarrah, Mehrdad Dianati, Saber Fallah, David Oxtoby, and Alex Mouzakitis. A survey on 3d object detection methods for autonomous driving applications. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3782–3795, 2019.

[7] Daniel Barath and Jiří Matas. Graph-cut ransac. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6733–6741, 2018.

[8] Luke Blades, Roy Douglas, Juliana Early, Chun Yi Lo, and Robert Best. Advanced driver-assistance systems for city bus applications. Technical report, SAE Technical Paper, 2020.

[9] Mariusz Bojarski, Anna Choromanska, Krzysztof Choromanski, Bernhard Firner, Larry J. Ackel, Urs Muller, Phil Yeres, and Karol Zieba. Visualbackprop: Efficient visualization of cnns for autonomous driving. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4701–4708. IEEE, 2018. ISBN 1538630818.

[10] M. Branch, T. Coleman, and Y. Li. A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems. *SIAM J. Sci. Comput.*, 21:1–23, 1999.

[11] C. G. BROYDEN. The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations. *IMA Journal of Applied Mathematics*, 6(1):76–90, 03 1970. ISSN 0272-4960. doi: 10.1093/imamat/6.1.76.

[12] Yingfeng Cai, Tianyu Luan, Hongbo Gao, Hai Wang, Long Chen, Yicheng Li, Miguel Angel Sotelo, and Zhixiong Li. Yolov4-5d: An effective and efficient object detector for autonomous driving. *IEEE Transactions on Instrumentation and Measurement*, 70:1–13, 2021. doi: 10.1109/TIM.2021.3065438.

[13] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1302–1310, 2017. doi: 10.1109/CVPR.2017.143.

[14] Florian Chabot, Mohamed Chaouch, Jaonary Rabarisoa, Celine Teuliere, and Thierry Chateau. Deep MANTA: A coarse-to-fine many-task network for joint 2D and 3D vehicle analysis from monocular image. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2040–2049, 2017.

[15] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[16] Leopoldo Gonzalez Clarembaux, Joshué Pérez, David Gonzalez, and Fawzi Nashashibi. Perception and control strategies for autonomous docking for electric freight vehicles. *Transportation Research Procedia*, 14:1516–1522, 2016. Transport Research Arena TRA2016.

[17] Solaris Bus & Coach. https://www.solarisbus.com. Accessed: 01.09.2024.

[18] Qi Dang, Jianqin Yin, Bin Wang, and Wenqing Zheng. Deep learning based 2d human pose estimation: A survey. *Tsinghua Science and Technology*, 24(6):663–676, 2019.

[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[20] Tommaso Dreossi, Shromona Ghosh, Xiangyu Yue, Kurt Keutzer, Alberto L. Sangiovanni-Vincentelli, and Sanjit A. Seshia. Counterexample-guided data augmentation. In *International Joint Conference on Artificial Intelligence*, 2018.

[21] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6568–6577. IEEE, 2019. ISBN 9781728148038.

[22] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24:381–395, 1981.

[23] R. Fletcher. *Practical Methods of Optimization*. Wiley, 2013. ISBN 9781118723180.

[24] Kui Fu, Jiansheng Peng, Qiwen He, and Hanxiao Zhang. Single image 3d object reconstruction based on deep learning: A review. *Multimedia Tools and Applications*, 80(1): 463–498, 2021.

[25] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):930–943, 2003. doi: 10.1109/TPAMI.2003.1217599.

[26] C.F. Gauss. *Theoria motus corporum coelestium in sectionibus conicis solem ambientium*. Carl Friedrich Gauss Werke. Sumtibus F. Perthes et I.H. Besser, 1809.

[27] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[28] Shromona Ghosh, Yash Vardhan Pant, Hadi Ravanbakhsh, and Sanjit A Seshia. Counterexample-guided synthesis of perception models and control. In *2021 American Control Conference (ACC)*, pages 3447–3454. IEEE, 2021.

[29] Ross Girshick. Fast r-cnn. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448. IEEE, 2015. ISBN 1467383910.

[30] Liang Gong, Yingxin Wu, Bishu Gao, Yefeng Sun, Xinyi Le, and Chengliang Liu. Real-time dynamic planning and tracking control of auto-docking for efficient wireless charging. *IEEE Transactions on Intelligent Vehicles*, 8(3):2123–2134, 2023. doi: 10.1109/TIV.2022.3189511.

[31] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of field robotics*, 37(3):362–386, 2020.

[32] Xian-Feng Han, Hamid Laga, and Mohammed Bennamoun. Image-based 3d object reconstruction: State-of-the-art and trends in the deep learning era. *IEEE transactions on pattern analysis and machine intelligence*, 43(5):1578–1604, 2019.

[33] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[34] Sabera Hoque, Shuxiang Xu, Ananda Maiti, Yuchen Wei, and Md. Yasir Arafat. Deep learning for 6d pose estimation of objects a case study for autonomous driving. *Expert Systems with Applications*, 223:119838, 2023. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2023.119838.

[35] Junjie Huang, Zheng Zhu, Feng Guo, and Guan Huang. The devil is in the details: Delving into unbiased data processing for human pose estimation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5699–5708, 2020. doi: 10.1109/CVPR42600.2020.00574.

[36] Xinyu Huang, Peng Wang, Xinjing Cheng, Dingfu Zhou, Qichuan Geng, and Ruigang Yang. The apolloscape open dataset for autonomous driving and its application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2702–2719, 2020. doi: 10.1109/TPAMI.2019.2926463.

[37] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7122–7131, 2017.

[38] Lei Ke, Shichao Li, Yanan Sun, Yu-Wing Tai, and Chi-Keung Tang. Gsnet: Joint vehicle pose and shape reconstruction with geometrical and scene-aware supervision. In *European Conference on Computer Vision*, 2020.

[39] Esraa Khatab, Ahmed Onsy, Martin Varley, and Ahmed Abouelfarag. Vulnerable objects detection for autonomous driving: A review. *Integration*, 78:36–48, 2021. ISSN 0167-9260. doi: https://doi.org/10.1016/j.vlsi.2021.01.002.

[40] Jinkyu Kim and John Canny. Interpretable learning for self-driving cars by visualizing causal attention. In *2017 IEEE International Conference on Computer Vision (ICCV)*, volume 2017-, pages 2961–2969. IEEE, 2017. ISBN 9781538610329.

[41] Jung Uk Kim, Sungjune Park, and Yong Man Ro. Towards human-like interpretable object detection via spatial relation encoding. In *2020 IEEE International Conference on Image Processing (ICIP)*, volume 2020-, pages 3284–3288. IEEE, 2020. ISBN 9781728163956.

[42] Kwang-Ju Kim, Pyong-Kun Kim, Yun-Su Chung, and Doo-Hyun Choi. Multi-scale detector for accurate vehicle detection in traffic surveillance data. *IEEE Access*, 7:78311–78319, 2019. doi: 10.1109/ACCESS.2019.2922479.

[43] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

[44] Laurent Kneip, Davide Scaramuzza, and Roland Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *CVPR 2011*, pages 2969–2976, 2011. doi: 10.1109/CVPR.2011.5995464.

[45] Yeongmin Ko, Younkwan Lee, Shoaib Azam, Farzeen Munir, Moongu Jeon, and Witold Pedrycz. Key points estimation and point instance segmentation approach for lane detection. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):8949–8958, 2021.

[46] Jeamin Koo, Jungsuk Kwac, Wendy Ju, Martin Steinert, Larry Leifer, and Clifford Nass. Why did my car just do that? explaining semi-autonomous driving actions to improve driver understanding, trust, and performance. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 9:269–275, 2015.

[47] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. Openpifpaf: Composite fields for semantic keypoint detection and spatio-temporal association. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):13498–13511, 2022. doi: 10.1109/TITS.2021.3124981.

[48] V. K. Kukkala, J. Tunnell, S. Pasricha, and T. Bradley. Advanced driver-assistance systems: A path toward autonomous vehicles. *IEEE Consumer Electronics Magazine*, 7(5): 18–25, 2018.

[49] Abhinav Kumar, Tim K. Marks, Wenxuan Mou, Chen Feng, and Xiaoming Liu. Uglli face alignment: Estimating uncertainty with gaussian log-likelihood loss. In *IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 778–782, 2019.

[50] Abhijit Kundu, Yin Li, and James M. Rehg. 3D-RCNN: Instance-level 3D object reconstruction via render-and-compare. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3559–3568, 2018.

[51] Abhijit Kundu, Yin Li, and James M. Rehg. 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3559–3568, 2018.

[52] Hyo-Jun Lee, Hanul Kim, Su-Min Choi, Seong-Gyun Jeong, and Yeong Jun Koh. Baam: Monocular 3d pose and shape reconstruction with bi-contextual attention module and attention-guided modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9011–9020, June 2023.

[53] Hyo-Jun Lee, Hanul Kim, Su-Min Choi, Seong-Gyun Jeong, and Yeong Jun Koh. Baam: Monocular 3d pose and shape reconstruction with bi-contextual attention module and attention-guided modeling. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9011–9020, 2023.

[54] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Ep n p: An accurate o (n) solution to the p n p problem. *International journal of computer vision*, 81:155–166, 2009.

[55] Shichao Li, Zengqiang Yan, Hongyang Li, and Kwang-Ting Cheng. Exploring intermediate representation for monocular vehicle pose estimation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1873–1883, 2020.

[56] Shichao Li, Zengqiang Yan, Hongyang Li, and Kwang-Ting Cheng. Exploring intermediate representation for monocular vehicle pose estimation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1873–1883, 2021. doi: 10.1109/CVPR46437.2021.00191.

[57] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1.

[58] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[59] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002. IEEE, 2021. ISBN 9781665428125.

[60] Xiao Xin Lu. A review of solutions for perspective-n-point problem in camera pose estimation. *Journal of Physics: Conference Series*, 1087:052009, 2018.

[61] R. C. Luo, C. T. Liao, K. L. Su, and K. C. Lin. Automatic docking and recharging system for autonomous security robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2953–2958, 2005.

[62] Karttikeya Mangalam, Ehsan Adeli, Kuan-Hui Lee, Adrien Gaidon, and Juan Carlos Niebles. Disentangling human dynamics for pedestrian locomotion forecasting with noisy supervision. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2784–2793, 2020.

[63] Weian Mao, Yongtao Ge, Chunhua Shen, Zhi Tian, Xinlong Wang, Zhibin Wang, and Anton van den Hengel. Poseur: Direct human pose regression withătransformers. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 72–88, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-20068-7.

[64] Mercedes-Benz Group. Testing of level 4 automated driving in beijing, 2024. URL https://group.mercedes-benz.com/innovations/product-innovation/autonomous-driving/level-4-beijing.html. Accessed: 2024-09-16.

[65] M. M. Michaek, B. Patkowski, and T. Gawron. Modular kinematic modelling of articulated buses. *IEEE Transactions on Vehicular Technology*, 69(8):8381–8394, 2020.

[66] Justinas Miseikis, Matthias Ruther, Bernhard Walzel, Mario Hirz, and Helmut Brunner. 3d vision guided robotic charging station for electric and plug-in hybrid vehicles, 2017.

[67] MMPose. Openmmlab pose estimation toolbox and benchmark. https://github.com/open-mmlab/mmpose, 2020.

[68] Tomas Möller and Ben Trumbore. Fast, minimum storage ray-triangle intersection. *Journal of Graphics Tools*, 2(1):2128, 1997.

[69] W. D. Montgomery, R. Mudge, E. L. Groshen, S. Helper, J. P. MacDuffie, and C. Carson. America's workforce and the self-driving future: Realizing productivity gains and spurring economic growth, 2018.

[70] Jorge J. Moré. The levenberg-marquardt algorithm: Implementation and theory. In G. A. Watson, editor, *Numerical Analysis*, pages 105–116, Berlin, Heidelberg, 1978. Springer Berlin Heidelberg. ISBN 978-3-540-35972-2.

[71] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3d bounding box estimation using deep learning and geometry. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5632–5640, 2016.

[72] J. Krishna Murthy, G.V. Sai Krishna, Falak Chhaya, and K. Madhava Krishna. Reconstructing vehicles from a single image: Shape priors for road scene understanding. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 724–731, 2017. doi: 10.1109/ICRA.2017.7989089.

[73] John A. Nelder and Roger Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

[74] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 483–499, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46484-8.

[75] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 427–436, 2015. doi: 10.1109/CVPR.2015.7298640.

[76] Tomasz Nowak and Piotr Skrzypczyski. Geometry-aware keypoint network: Accurate prediction of point features in challenging scenario. In *17th Conference on Computer Science and Intelligence Systems (FedCSIS)*, pages 191–200, 2022.

[77] Shiye Pan and Xinmei Wang. A survey on perspective-n-point problem. In *2021 40th Chinese Control Conference (CCC)*, pages 2396–2401, 2021. doi: 10.23919/CCC52363.2021.9549863.

[78] George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin Murphy. Towards accurate multi-person pose estimation in the wild, 2017.

[79] P. Petrov, C. Boussard, S. Ammoun, and F. Nashashibi. A hybrid control for automatic docking of electric vehicles for recharging. In *IEEE International Conference on Robotics and Automation*, pages 2966–2971, 2012.

[80] Dailys Arronde Pérez, Harald Gietler, and Hubert Zangl. Automatic uncertainty propagation based on the unscented transform. In *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pages 1–6, 2020.

[81] Manikandasriram Srinivasan Ramanagopal, Cyrus Anderson, Ram Vasudevan, and Matthew Johnson-Roberson. Failing to learn: Autonomously identifying perception failures for self-driving cars. *IEEE robotics and automation letters*, 3(4):3860–3867, 2018. ISSN 2377-3766.

[82] N. Dinesh Reddy, Minh Vo, and Srinivasa G. Narasimhan. Occlusion-net: 2d/3d occluded keypoint localization using graph networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7318–7327, 2019. doi: 10.1109/CVPR.2019.00750.

[83] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525, 2016.

[84] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018.

[85] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016. doi: 10.1109/CVPR.2016.91.

[86] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[87] Helge Rhodin, Mathieu Salzmann, and Pascal Fua. Unsupervised geometry-aware representation for 3d human pose estimation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 750–767, 2018.

[88] Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. In *International Conference on Learning Representations*, 2021.

[89] Schunk Carbon Technology. Schunk smart charging, 2021. URL https://www.schunk-carbontechnology.com/en/smart-charging.

[90] Sina Shafaei, Stefan Kugele, Mohd Hafeez Osman, and Alois Knoll. Uncertainty in machine learning: A safety perspective on autonomous driving. In *Computer Safety, Reliability, and Security: SAFECOMP 2018 Workshops, ASSURE, DECSoS, SASSUR, STRIVE, and WAISE, Västerås, Sweden, September 18, 2018, Proceedings 37*, pages 458–464. Springer, 2018.

[91] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439, 2019.

[92] Sehajbir Singh and Baljit Singh Saini. Autonomous cars: Recent developments, challenges, and possible solutions. In *IOP conference series: Materials science and engineering*, volume 1022, page 012028. IOP Publishing, 2021.

[93] Xibin Song, Peng Wang, Dingfu Zhou, Rui Zhu, Guan Chenye, Yuchao Dai, Hao Su, Hongdong li, and Ruigang Yang. ApolloCar3D: A large 3D car instance understanding benchmark for autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5447–5457, 06 2019.

[94] Xiao Sun, Jiaxiang Shang, Shuang Liang, and Yichen Wei. Compositional human pose regression. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2621–2630, 2017. doi: 10.1109/ICCV.2017.284.

[95] Héctor Corrales Sánchez, Antonio Hernández Martínez, Rubén Izquierdo Gonzalo, Noelia Hernández Parra, Ignacio Parra Alonso, and David Fernández-Llorca. Simple baseline for vehicle pose estimation: Experimental validation. *IEEE Access*, 8:132539–132550, 2020. doi: 10.1109/ACCESS.2020.3010307.

[96] Ali Taghibakhshi, Nathan Ogden, and Matthew West. Local navigation and docking of an autonomous robot mower using reinforcement learning and computer vision. *2021 13th International Conference on Computer and Automation Engineering (ICCAE)*, pages 10–14, 2021.

[97] Zhi Tian, Hao Chen, and Chunhua Shen. Directpose: Direct end-to-end multi-person pose estimation, 2019.

[98] TIER IV, Inc. Tier iv unveils l4 ride to propel autonomous bus services across japan. TIER IV, Inc. Media Release, August 2024. URL https://tier4.jp/en/media/detail/?sys_id=78IgAImD8PzSvryYmEy7zt. Accessed: 2024-09-16.

[99] Mukhiddin Toshpulatov, Wookey Lee, Suan Lee, and Arousha Haghighian Roudsari. Human pose, hand and mesh estimation using deep learning: a survey. *The Journal of Supercomputing*, 78(6):7616–7654, 2022. doi: https://doi.org/10.1007/s11227-021-04184-7.

[100] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[101] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018.

[102] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

[103] VUMO. https://vumo.ai/. Accessed: 16.09.2024.

[104] Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, and Guiguang Ding. Yolov10: Real-time end-to-end object detection. *arXiv preprint arXiv:2405.14458*, 2024.

[105] Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16611–16621, 2021.

[106] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-CAM: score-weighted visual explanations for convolutional neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2020.

[107] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, D. Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43:3349–3364, 2021.

[108] Ke Wang, Yong Wang, Bingjun Liu, and Junlan Chen. Quantification of uncertainty and its applications to complex domain for autonomous vehicles perception system. *IEEE Transactions on Instrumentation and Measurement*, 72:1–17, 2023.

[109] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, W. Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *European Conference on Computer Vision*, 2018.

[110] Li-Hua Wen and Kang-Hyun Jo. Deep learning-based perception systems for autonomous driving: A comprehensive survey. *Neurocomputing*, 489:255–270, 2022. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2021.08.155.

[111] Xin Wen, Junsheng Zhou, Yu-Shen Liu, Hua Su, Zhen Dong, and Zhizhong Han. 3d shape reconstruction from 2d images with disentangled attribute flow. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3793–3803, 2022. doi: 10.1109/CVPR52688.2022.00378.

[112] Bichen Wu, Alvin Wan, Forrest Iandola, Peter H. Jin, and Kurt Keutzer. Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 446–454, 2017. doi: 10.1109/CVPRW.2017.60.

[113] Di Wu, Zhaoyong Zhuang, Canqun Xiang, Wenbin Zou, and Xia Li. 6d-vnet: End-to-end 6dof vehicle pose estimation from monocular rgb images. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1238–1247, 2019.

[114] Tianfu Wu and Xi Song. Towards interpretable object detection by unfolding latent structures. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6032–6042, 2019. doi: 10.1109/ICCV.2019.00613.

[115] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32:4–24, 2019.

[116] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2014.

[117] Liu Yang, Steve Hanneke, and Jaime Carbonell. A theory of transfer learning with applications to active learning. *MACHINE LEARNING*, 90(2):161–189, FEB 2013. ISSN 0885-6125. doi: 10.1007/s10994-012-5310-y.

[118] Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. In S Bengio, H Wallach, H Larochelle, K Grauman, N CesaBianchi, and R Garnett, editors, *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 31 (NIPS 2018)*, volume 31 of *Advances in Neural Information Processing Systems*, 2018. 32nd Conference on Neural Information Processing Systems (NIPS), Montreal, CANADA, DEC 02-08, 2018.

[119] Cui Youjing and Sam Ge Shuzhi. Autonomous vehicle positioning with GPS in urban canyon environments. *IEEE Transactions on Robotics and Automation*, 19(1):15–25, 2003.

[120] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE Access*, 8:58443–58469, 2020. doi: 10.1109/ACCESS.2020.2983149.

[121] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 818–833, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10590-1.

[122] Feng Zhang, Xiatian Zhu, Hanbin Dai, Mao Ye, and Ce Zhu. Distribution-aware coordinate representation for human pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[123] Haotian Zhang, Haorui Ji, Aotian Zheng, Jenq-Neng Hwang, and Ren-Hung Hwang. Monocular 3d localization of vehicles in road scenes. In *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 2855–2864, 2021. doi: 10.1109/ICCVW54120.2021.00320.

[124] Yu Zhang, Peter Tiňo, Aleš Leonardis, and Ke Tang. A survey on neural network interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 5(5): 726–742, 2021.

[125] Ce Zheng, Wenhan Wu, Chen Chen, Taojiannan Yang, Sijie Zhu, Ju Shen, Nasser Kehtarnavaz, and Mubarak Shah. Deep learning-based human pose estimation: A survey. *ACM Computing Surveys*, 56(1):1–37, 2023.