3DCMA : 3D Convolution with Masked Attention for Ego Vehicle Speed Estimation

Athul M. Mathew Thariq Khalid Riad Souissi
Elm Company
Riyadh, Saudi Arabia
{amathew, tkadavil, rsouissi}@elm.sa

Abstract

Speed estimation of an ego vehicle is crucial to enable autonomous driving and advanced driver assistance technologies. Due to functional and legacy issues, conventional methods depend on in-car sensors to extract vehicle speed through the Controller Area Network (CAN) bus. However, it is desirable to have modular systems that are not susceptible to external sensors to execute perception tasks. In this paper, we propose a novel 3D-CNN with masked-attention (3DCMA) architecture to estimate ego vehicle speed using a single front-facing monocular camera. To demonstrate the effectiveness of our method, we conduct experiments on two publicly available datasets, nuImages and KITTI. We additionally introduce a synthetic dataset for ego vehicle speed estimation (SEVS dataset) that bridges the gaps in the existing real-world datasets. Our method outperforms the current state-of-the-art architecture for video vision by 27% and 34% in nuImages and KITTI datasets, respectively. We also demonstrate masked-attention's efficacy by comparing our method with a traditional 3D-CNN. Our method achieved an error reduction of 23% and 25% for the datasets mentioned above when compared against 3D-CNN without masked-attention.

1. Introduction

The impact of electric vehicles today in contributing to an energy-efficient and sustainable world is immense [19]. It is a significant influencing factor in the global push against climate change. To this end, self-driving vehicles add further value in enabling smart mobility, planning, and control for intelligent transportation systems. According to [31], predicting the ego vehicle speed reduces fuel consumption and optimizes cruise control [30].

Autonomous cars use stereo cameras, LiDAR (Light Detection and Ranging), and radars to estimate the speed of the ego vehicle and other vehicles. The combination of these sensors gives great accuracy leading to meticulous navigation to avoid crashes and ensure the safety of the vehicle and its surrounding. Camera-LiDAR fusion exploits the video stream and 3D point clouds simultaneously to get depth information of objects around the vehicle. Thus, the vehicle speed can be estimated from the multitude of features learnt using multiple sensor modalities. However, such multi-sensor dependant systems are not cost effective.

Figure 1. Estimation of ego-vehicle speed (green) using a continuous camera stream

So far, very little work has been done on speed estimation of a moving car using monocular camera. An example of ego-vehicle speed estimation using camera stream is shown in Fig. 1. In our work, we present a 3D Convolutional Neural Network (3D-CNN) architecture trained on short videos using their grayscale image frames and the corresponding lane line segmentation masks. Using our neural network architecture, we are able to estimate the speed of the ego vehicle, which can, in turn help to estimate the speed of vehicles of interest (VOI) in the surrounding environment.

Most players in the autonomous driving industry rely heavily on thousands of hours of manual driving data. Most synthetic datasets such as those tabulated in [23] provide

data from multiple sensors such as cameras, LiDAR, GPS, and IMU. In this paper, we introduce a synthetic dataset called **S**ynthetic **E**go **V**ehicle **S**peed (**SEVS**) dataset, that comprises of car-mounted front-facing video streams with vehicle speed ground truth generated under different environmental conditions with varying road textures and lane markers. We believe that our work is effective yet simple and can be helpful as modular components in autonomous or intelligent traffic systems.

2. Related Work

The work done by [38] is one of the early works to estimate the ego-motion using correspondence points detection, road region detection, moving object detection, and other derived features. Furthermore, 8-point algorithm [16] and RANSAC [9] are applied to get the essential matrix of ego-motion. The recent work in [3] implemented an end-to-end CNN-LSTM network to estimate the speed of an ego vehicle. The work performs evaluation on DBNet [5] and comma.ai speed challenge dataset [1]. Other works, such as [25], propose speed estimation of vehicles from a CCTV point of view. Most require camera calibration and fixed view so that the vehicles pass through certain lines or regions of interest. FlowNet [20] and PWC-Net [32] are deep neural networks to estimate optical flow in videos. Further research in [17, 28] make use of FlowNet or PWC-Net to estimate the ego vehicle speed. However, they perform ego vehicle speed estimation by further post-processing on the optical flow pixel velocity. None of the works demonstrate the end-to-end architecture capability where the speed could be learned with differentiation of the loss function.

2.1. 3D Convolutional Neural Network

3D Convolutional Neural Networks are the best in learning spatio-temporal features and thus help in video classification [22], human action recognition [21], and sign language recognition [27]. There have been recent works [13,35,36] which use attention on top of 3D-CNN; however, they are limited to action recognition use cases. Very few works such as [7,10,15] perform regression using 3D-CNNs however they perform spatial localization-related tasks such as human pose or 3D hand pose. Our work performs the regression across the spatio-temporal aspects by having the 3D-CNN attend to both the visual features of the gray images and the lane masks.

2.2. Vision Transformers

Transformer models for video have been proposed in a plethora of architectures. The Video Transformer architectures can be classified based on the embeddings (backbone and minimal embeddings), tokenization (patch tokenization, frame tokenization, clip tokenization), and positional embeddings. We chose the Video Vision Transformer(ViViT) [2] for our experiments due to its representation of the 3D convolution in the form of Tubelet embedding as seen in Fig. 2. ViViT is easily reproducible and has a good balance between the parameters and accuracy for small datasets. Moreover, ViViT-H scores an accuracy of 95.8, just below the 95.9 accuracy score by Swin-L as per the Video Transformers Survey [29] over HowTo100M [26].

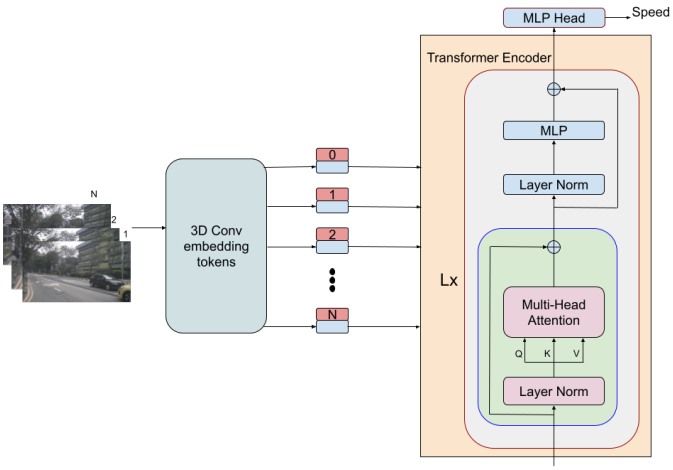


Figure 2. Architecture of ViViT. Here the frames from the video(N) are tokenized using 3D-Convolutional tubelet embeddings and further passed to multiple transformer encoders to regress the speed value finally. The Transformer Encoder is trained with the spatio-temporal embeddings

3. Methods

We aim to estimate the ego vehicle speed by relying purely on video streams from a monocular camera. The authors of [33] have proved the capability of a 3D-CNN to learn spatio-temporal features.

A 2D convolution operation over an image I using a kernel K of size m is given by [14] as :

$$S(i,j) = (I * K)(i,j) \tag{1}$$

$$= \sum_m \sum_n I(i,j)K(i-m, j-n) \tag{2}$$

Expanding further on the above equation, the 3D convolution operation can be expressed as :

$$S(h,i,j) = (I * K)(h,i,j) \tag{3}$$

$$= \sum_l \sum_m \sum_n I(h,i,j)K(h-l, i-m, j-n) \tag{4}$$

where h is the additional dimension that includes the number of frames the kernel has to go through. Here the

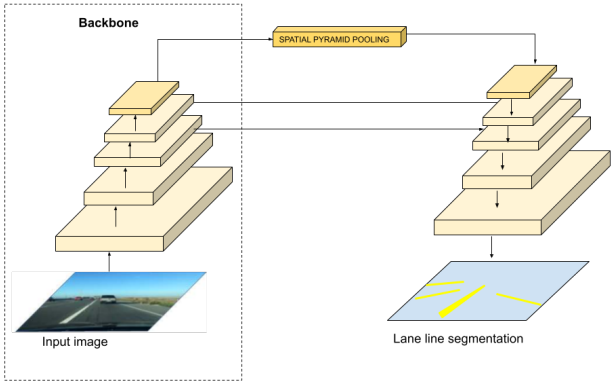


Figure 3. Architecture from [37] modified for Lane line segmentation comprises of an encoder and a decoder

kernel is convoluted with the concatenation of the grayscale images and lane line segmentation masks. To this extent, we incorporate a 3D-CNN network to preserve the temporal information of the input signals and compute the ego vehicle speed. 3D-CNNs can learn spatial and temporal features simultaneously using 3D kernels [21]. We use small receptive fields of $3 \times 3 \times 3$ as our convolutional kernels throughout the network. Many 3D-CNN architectures lose big chunks of temporal information after the first 3D pooling layer. We refer to the pooling kernel size as $d \times k \times k$, where d is the kernel temporal depth, and s is the spatial kernel size. Similar to [33], we used $d = 1$ for the first max pooling layer to preserve the temporal information. This way, we ensure that the temporal information does not collapse entirely after the initial convolutional layers. In this paper, our contribution includes adding a masked-attention layer into the 3D-CNN architecture to guide the model to focus on relevant features that help with ego-vehicle speed computation. We show that the error in speed estimation reduces by adding masked-attention to the 3D-CNN network. Further details about the impact of masked-attention are described as part of the ablation study in section 5.1.

3.1. Masked-Attention

Convolutional neural networks comprise a learned set of filters, where each filter extracts a different feature from the image [8]. We aim to inhibit or exhibit the activation of features based on the appearance of objects of interest in the images. Typical scenes captured by car-mounted imaging devices include background objects such as the sky, and environment vehicles, which do not contribute to ego-vehicle speed estimation. In fact, the relative motion of environmental vehicles often contributes negatively to the ability of the neural network to inhibit irrelevant features.

To inhibit and exhibit features based on relevance, we concatenate the masked-attention map to the input image before passing it through the neural network. We utilize a single-shot network with a shared encoder and three separate decoders that accomplish specific tasks such as object detection, drivable area segmentation, and lane line segmentation [37]. CSP-Darknet [34] is chosen as the backbone network of the encoder, while the neck is mainly composed of Spatial Pyramid Pooling (SPP) module [18] and Feature Pyramid Network (FPN) module [24]. SPP generates and fuses features of different scales, and FPN fuses features at different semantic levels, making the generated features contain multiple scales and semantic level information.

The masked-attention map is generated from input video sequences using the lane line segmentation branch. The concatenation of lane segmentation as an additional channel to the camera channel allows the 3D-CNN to focus on the apparent displacement of the lane line segments in the video sequences to best estimate the ego-vehicle speed. Fig. 3 shows the modified architecture designed by [37] for extraction of lane line segments.

3.2. Network Architecture

Our 3D-CNN architecture with masked-attention for ego vehicle speed estimation is illustrated in Fig. 4.

We convert the RGB stream to grayscale since color information is not vital for speed estimation. However, masked-attention map is concatenated as an additional channel to the grayscale image. To reduce the computational complexity and memory requirement, the original input streams are resized to 64×64 before feeding them into the network. Thus the input to the model has a dimension of $n \times 64 \times 64 \times 2$, where n is the number of frames in the temporal sequence. All convolutional 3D layers use a fixed kernel size of $3 \times 3 \times 3$ as recommended in [33]. The initial pooling layer uses a kernel size of $1 \times 2 \times 2$ to preserve the temporal information. The subsequent pooling layer, which appears at the center of the network, compresses the temporal and spatial domains with a kernel size of $2 \times 2 \times 2$. We incorporate six 3D convolutional layers with the number of filters for the layers from $1-6$ being $32, 32, 64, 64, 128, 128$ respectively. Finally, four fully connected layers have $512, 256, 64$ and 1 nodes.

The L2 loss function which we used for the 3D-CNN can be described as :

$$\mathscr{L}_{speed} = \frac{1}{n}\sum_{i=0}^{n}(S_i - \hat{S}_i)^2 \tag{5}$$

$$= \frac{1}{n}\sum_{i=0}^{n}(S_i - W^T X)^2 \tag{6}$$

$$= \frac{1}{n}\sum_{i=0}^{n}(S_i - W^T (X_I + X_M))^2 \tag{7}$$

where n is the number of frames in the input and S_i is the

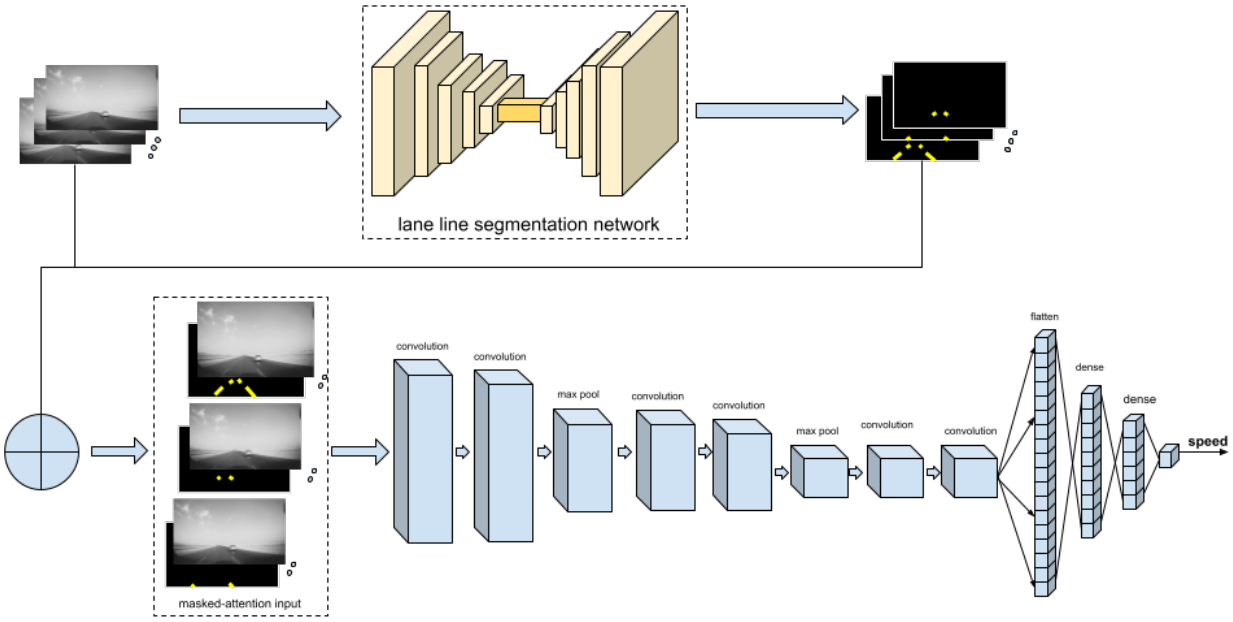


Figure 4. architecture of 3DCMA

speed value ground truth of ith corresponding frame, and $\hat{S}_i$ is the inferred speed value. X_I is the grayscale image channel, and X_M is the masked-attention channel for every frame. W is the weight tensor of the 3D convolutional kernel.

4. Experimentation

4.1. Datasets

In this paper, we utilized two public datasets for our experiments - nuImages and KITTI. Some sample images extracted from video sequences for nuImages and KITTI are shown in Fig. 5. In addition to the publicly available datasets, we introduce a synthetic dataset for ego-vehicle speed estimation having vehicular speeds simulated in ranges of 0-120 km/hr.

4.1.1 nuImages Dataset

nuImages is derived from nuScenes [4], and it is a large-scale autonomous driving dataset having 93k video clips of 6 seconds each. Each video clip consists of 13 frames spaced out at 2 Hz. The annotated images include rain, snow, and night time, which are important for autonomous driving applications. The vehicle speed is extracted from the CAN bus data and linked to the sample data through sample tokens.

4.1.2 KITTI Dataset

The KITTI Vision Benchmark Suite [11, 12] is a public dataset containing raw data recordings that are captured and synchronized at 10 Hz. We utilized the RGB stream extracted from camera ID 03 only. The ego-vehicle speed values are extracted from IMU sensor readings.We used data from City and Road categories. To facilitate future benchmarks from the research community, we report our train and test splits in Table 1 as well.

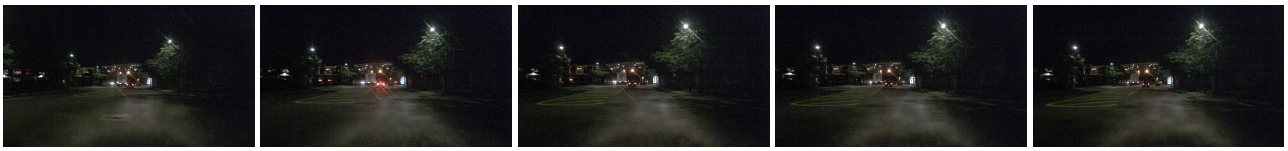
KITTI Category	Train		Test
City	2011_09_26_drive_	0002, 0005, 0009 0011, 0013, 0014 0048, 0051, 0056 0059, 0084, 0091 0095, 0096, 0104 0106, 0113	0001 0117
	2011_09_28_drive_	0001	
	2011_09_29_drive_	0071	
Road	2011_09_26_drive_	0015, 0027, 0028 0029, 0032, 0052	0070 0101
	2011_09_29_drive_	0004, 0016, 0042 0047	

Table 1. Train and Test video samples for KITTI dataset

(a) Road category images from KITTI dataset

(b) City category images from KITTI dataset

(c) Daytime images from nuImages dataset



(d) Night time images from nuImages dataset

Figure 5. Visualization of sample images - KITTI and nuImages dataset

4.1.3 Synthetic Ego Vehicle Speed (SEVS) dataset

We rendered the synthetic simulations using Blender [6] 3D modeling tool. Road length was modeled for a distance of 1000 meters. Cat-eye and solid lanes were modeled to increase the variations within the dataset. We utilized *Rigacar* addon to simulate vehicle speeds in a controlled manner. We additionally mounted the front-facing cameras at different heights within the car and introduced focal-length perturbations to increase the data diversity. The dataset is also complemented with variations in environmental conditions with the inclusion of normal, sunny, sandstorm, and night simulations. Some sample images from SEVS is shown in Fig. 6. We generated video sequences mimicking a car-mounted dash-cam for car speeds ranging from 0-120 km/hr.

4.2. Evaluation metrics

We utilize the conventional evaluation protocol that is used in the literature for the task of regression - Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) [46].

We compute the MAE and RMSE as follows :

$$RMSE = \sqrt{(\frac{1}{n})\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{8}$$

$$MAE = (\frac{1}{n})\sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{9}$$

where y_i denotes the ground truth ego-vehicle speed value and $\hat{y}_i$ denotes the predicted speed value by the AI model.

5. Results

We evaluate the performance of our proposed 3DCMA architecture and compare it against the standard ViViT with spatio-temporal attention. We report the evaluation scores on the test set for KITTI, nuImages and SEVS datasets in Table 2.

		Evaluation Metric	
Dataset	*Method*	*RMSE*	*MAE*
nuImages	ViViT	1.782	1.326
	3DCMA	**1.297**	**0.974**
KITTI	ViViT	5.024	4.324
	3DCMA	**3.290**	**2.528**
SEVS	ViViT	1.639	0.506
	3DCMA	**0.982**	**0.506**

Table 2. Evaluation on test datasets for (a)ViViT (b)3DCMA

We observed approximately 27% improvement in RMSE and MAE for 3DCMA compared to ViViT for the nuImages dataset, while the improvement in RMSE and MAE on the KITTI dataset was 34.5% and 41.5% respectively. Finally, we observed 40% improvement in RMSE for 3DCMA compared to ViViT on the SEVS dataset.

5.1. Ablation Study

To further understand the importance of masked-attention, we conducted an ablation study by removing masked-attention input to the 3D-CNN network. It is to

(a) Variations in lane marker type and road texture. The left image shows dotted lanes, whereas the right image shows linear painted lanes

(b) Variation in environment condition. The left image shows the driving condition under sunlight. The middle image shows the sand storm condition in desert areas, and the right image shows night synthetic driving conditions

Figure 6. Visualization of sample images - SEVS dataset

be noted that the input to the 3D-CNN model is a single-channel grayscale image after the removal of the masked-attention input. Evaluation scores for the test datasets are shown in Table 3. The addition of masked-attention reduced RMSE by 23.6% and MAE by 25.9% for the nuImages dataset, while the reduction in RMSE and MAE were 25.8% and 30.1% for the KITTI dataset.

		Evaluation Metric	
Dataset	*Method*	*RMSE*	*MAE*
nuImages	3D-CNN without MA	1.698	1.315
	3DCMA	**1.297**	**0.974**
KITTI	3D-CNN without MA	4.437	3.617
	3DCMA	**3.290**	**2.528**

Table 3. Evaluation on test datasets for (a)3D-CNN without masked-attention (b)3DCMA

6. Discussion

In this paper, we propose a modified 3D-CNN architecture with masked-attention employed for ego vehicle speed estimation using single-camera video streams. We evaluated the performance of our proposed architecture on two publicly available datasets - nuImages and KITTI. We compared our proposed method against a recent state-of-the-art transformer network for videos, ViViT. We additionally investigated the impact of employing masked-attention to 3D-CNN and saw that the injection of masked-attention improved the MAE and RMSE scores across all scenarios.

One limiting factor we noticed in these datasets is the lack of driving data for higher vehicle speeds. The vehicle speeds are available only up to $20m/s$, thus limiting the scope of deploying these models for highway driving scenarios. To overcome this bottleneck, we generated synthetic dataset for covering wider ranges of driving speeds.

7. Conclusion

Though ViViTs can model long-range interactions across videos right from the first layer, we demonstrated that a 3D-CNN injected with masked-attention performed better overall across all test scenarios. In this paper, we introduced a simple yet effective 3D-CNN with masked-attention architecture that can effectively compute the ego-vehicle speed using monocular camera streams. We additionally introduced a synthetic dataset that can contribute to current research. Immediate future work is the extension of current work to utilize the speed of ego vehicle to estimate the speeds and locations of environment vehicles for in-vehicle motion and path planning.

8. Acknowledgement

We express our gratitude to Muhammad AL-Qurishi and Arshad Khan for their review and feedback.

References

[1] comma.ai speed challege. https://github.com/commaai/speedchallenge, 2018.

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6836–6846, 2021.

[3] Hitesh Linganna Bandari and Binoy B Nair. An end to end learning based ego vehicle speed estimation system. In *2021 IEEE International Power and Renewable Energy Conference (IPRECON)*, pages 1–8. IEEE, 2021.

[4] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020.

[5] Yiping Chen, Jingkang Wang, Jonathan Li, Cewu Lu, Zhipeng Luo, Han Xue, and Cheng Wang. Lidar-video driving dataset: Learning driving policies effectively. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5870–5878, 2018.

[6] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[7] Xiaoming Deng, Shuo Yang, Yinda Zhang, Ping Tan, Liang Chang, and Hongan Wang. Hand3d: Hand pose estimation using 3d neural network. *arXiv preprint arXiv:1704.02224*, 2017.

[8] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning, 2016.

[9] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[10] Liuhao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. 3d convolutional neural networks for efficient and robust hand pose estimation from single depth images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1991–2000, 2017.

[11] A Geiger, P Lenz, C Stiller, and R Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[13] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 244–253, 2019.

[14] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, Cambridge, MA, USA, 2016. http://www.deeplearningbook.org.

[15] Agne Grinciunaite, Amogh Gudi, Emrah Tasli, and Marten den Uyl. Human pose estimation in space and time using 3d cnn. In *European Conference on Computer Vision*, pages 32–39. Springer, 2016.

[16] Richard I Hartley. In defense of the eight-point algorithm. *IEEE Transactions on pattern analysis and machine intelligence*, 19(6):580–593, 1997.

[17] Jun Hayakawa and Behzad Dariush. Ego-motion and surrounding vehicle state estimation using a monocular camera. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 2550–2556. IEEE, 2019.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *Computer Vision – ECCV 2014*, pages 346–361. Springer International Publishing, 2014.

[19] Graeme Hill, Oliver Heidrich, Felix Creutzig, and Phil Blythe. The role of electric vehicles in near-term mitigation pathways and achieving the uk's carbon budget. *Applied Energy*, 251:113111, 2019.

[20] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017.

[21] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2012.

[22] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[23] Andreas Kloukiniotis, Andreas Papandreou, Christos Anagnostopoulos, Aris Lalos, Petros Kapsalas, Duong-Van Nguyen, and Konstantinos Moustakas. Carlascenes: A synthetic dataset for odometry in autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4520–4528, June 2022.

[24] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection, 2016.

[25] Hector Mejia, Esteban Palomo, Ezequiel López-Rubio, Israel Pineda, and Rigoberto Fonseca. Vehicle speed estimation using computer vision and evolutionary camera calibration. In *NeurIPS 2021 Workshop LatinX in AI*, 2021.

[26] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2630–2640, 2019.

[27] Junfu Pu, Wengang Zhou, and Houqiang Li. Iterative alignment network for continuous sign language recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4165–4174, 2019.

[28] Róbert-Adrian Rill. Speed estimation evaluation on the kitti benchmark based on motion and monocular depth information. *arXiv preprint arXiv:1907.06989*, 2019.

[29] Javier Selva, Anders S Johansen, Sergio Escalera, Kamal Nasrollahi, Thomas B Moeslund, and Albert Clapés. Video transformers: A survey. *arXiv preprint arXiv:2201.05991*, 2022.

[30] Thomas Stanger and Luigi del Re. A model predictive cooperative adaptive cruise control approach. In *2013 American control conference*, pages 1374–1379. IEEE, 2013.

[31] Chao Sun, Xiaosong Hu, Scott J Moura, and Fengchun Sun. Velocity predictors for predictive energy management in hybrid electric vehicles. *IEEE Transactions on Control Systems Technology*, 23(3):1197–1204, 2014.

[32] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018.

[33] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks, 2014.

[34] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Scaled-yolov4: Scaling cross stage partial network, 2020.

[35] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[36] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 284–293, 2019.

[37] Dong Wu, Manwen Liao, Weitian Zhang, and Xinggang Wang. Yolop: You only look once for panoptic driving perception, 2021.

[38] Koichiro Yamaguchi, Takeo Kato, and Yoshiki Ninomiya. Vehicle ego-motion estimation and moving object detection using a monocular camera. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 4, pages 610–613. IEEE, 2006.