MultiNet++: Multi-Stream Feature Aggregation and Geometric Loss Strategy for Multi-Task Learning

Sumanth Chennupati[1,3], Ganesh Sistu[2], Senthil Yogamani[2] and Samir A Rawashdeh[3]
[1]Valeo North America, [2]Valeo Vision Systems, [3]University of Michigan-Dearborn
schenn@umich.edu, ganesh.sistu@valeo.com, senthil.yogamani@valeo.com, srawa@umich.edu

Abstract

Multi-task learning is commonly used in autonomous driving for solving various visual perception tasks. It offers significant benefits in terms of both performance and computational complexity. Current work on multi-task learning networks focus on processing a single input image and there is no known implementation of multi-task learning handling a sequence of images. In this work, we propose a multi-stream multi-task network to take advantage of using feature representations from preceding frames in a video sequence for joint learning of segmentation, depth, and motion. The weights of the current and previous encoder are shared so that features computed in the previous frame can be leveraged without additional computation. In addition, we propose to use the geometric mean of task losses as a better alternative to the weighted average of task losses. The proposed loss function facilitates better handling of the difference in convergence rates of different tasks. Experimental results on KITTI, Cityscapes and SYNTHIA datasets demonstrate that the proposed strategies outperform various existing multi-task learning solutions.

1. Introduction

Multi-task learning (MTL) [2] aims to jointly solve multiple tasks by leveraging the underlying similarities between independent or interdependent tasks. It is perceived as an attempt to improve generalization by learning a common feature representation for multiple tasks. Improvements in prediction accuracy and reduced computation complexities are significant benefits of MTL. This allowed deployment of MTL in various applications in computer vision (especially scene understanding) [55, 22, 4], natural language processing [43, 11], speech recognition [57, 50], reinforcement learning [9, 8], drug discovery [34, 25], *etc*.

MTL networks were mainly built using Convolution Neural Networks (CNNs). These networks were usually limited to operate on a single stream of input data. However, numerous works demonstrate using multiple streams of data as input to CNNs can improve performance drastically compared to using a single stream of input data. Recent attempts that use consecutive frames in a video sequence for semantic segmentation [46, 51, 48], activity recognition [19, 49], optical flow estimation [35], moving object detection [47, 56] are examples demonstrating the benefits of using multiple streams of input data. Similarly, a pair of images from stereo vision cameras [28] or multiple images from different cameras of a surround view system of a car can also be processed as multiple streams of input to CNNs. Some works considered processing input data from different domains [41] to solve certain tasks that require multi-modal data representations.

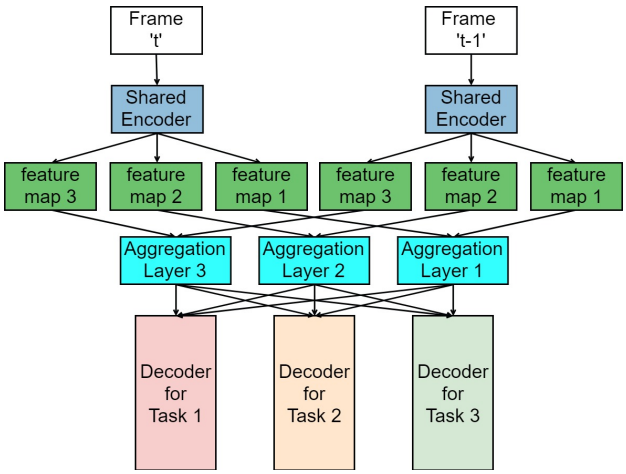


Figure 1: Illustration of MultiNet++ where feature aggregation is performed to combine intermediate output data obtained from a shared encoder that operates on multiple input streams (Frames 't' and 't-1'). The aggregated features are later processed by task specific decoders.

These significant benefits demand the construction of a multi-task learning network that can operate on multiple streams of input data. Thus, we propose MultiNet++, a novel multi-task network using simple feature aggrega-

978-1-7281-2506-0/19/$31.00 ©2019 IEEE
DOI 10.1109/CVPRW.2019.00159

tion methods as shown in Figure 1 to combine multiple streams of input data, which can be further processed by task-specific decoders. Figure 1 illustrates a generic way to aggregate features temporally and we make use of a simple summation junction to combine temporal features in our experiments. MultiNet++ would be ideal to process video sequences for tasks like semantic segmentation, depth estimation, optical flow estimation, object detection and tracking, *etc*. with improved efficiency. We also propose a novel loss strategy for multi-task learning based on geometric mean representation to prioritize learning of all tasks equally. The motivation for MultiNet++ is derived from our position paper NeurAll [52] which proposes to move towards a unified visual perception model for autonomous driving. We propose to use three diverse tasks namely segmentation, depth estimation and motion segmentation which make use of appearance, geometry and motion cues respectively.

The rest of the contents in this paper are structured as follows. Section 2 reviews related work using feature aggregation for multiple streams of inputs to CNNs and different task loss weighing strategies used in MTL. Section 3 discusses in detail the proposed MultiNet++ network along with the geometric loss strategy used in this paper. Section 4 presents the experimental results on automotive datasets mainly KITTI [12], Cityscapes [6] and SYNTHIA [39]. Finally, Section 5 summarizes the paper with key observations and concluding remarks.

2. Related Work

2.1. Multi-Task Learning

Multi-task learning typically consists of two blocks, shared parameters, and task-specific parameters. Shared parameters are learned to represent commonalities between several tasks while task-specific parameters are learned to perform independent processing. In MTL networks built using CNNs, shared parameters are called encoders as they perform the key feature extraction and the task-specific parameters are called decoders as they decode the information from encoders. MTL networks are classified into hard parameter sharing or soft parameter sharing categories based on how they share their parameters. In hard parameter sharing, initial layers or parameters are shared between different tasks such that these parameters are common for all tasks. In soft parameter sharing, different tasks are allowed to have different initial layers with some extent of sharing between them. Cross stitch [31] and sluice networks [40] are examples of soft parameter sharing. Majority of the works in MTL use hard parameter sharing as it is easier to build and computationally less complex.

The performance of the MTL network is highly dependent on their shared parameters as they contain the knowledge learned from different tasks [2, 1, 38]. Inappropriate learning of these parameters can induce biased representations for a particular task which can hurt the performance of MTL networks. This phenomenon is referred to as negative transfer learning. In order to prevent it, meaningful feature representations and balanced learning methods are required.

2.2. Feature Aggregation

Different outputs from initial or mid-level convolution layers from CNNs (referred to as extracted features) are forwarded to the next stage of processing using feature aggregation. Feature aggregation is a meaningful way to combine these extracted features. These features can be extracted from different CNNs operating on different input data [62, 37] or from a CNN operating on different resolutions of input [24]. Ranjan *et al*. [36] combines intermediate outputs from a CNN and passes to next stages of processing. Yu *et al*. [60] proposed several possibilities of feature aggregation.

There are plenty of choices to perform feature aggregation. These choices range from using simple concatenation techniques to complex Long Short Term Memory units (LSTMs) [17] or recurrent units. Simple concatenation or addition layers can capture short term temporal cues from a video sequence. Sun *et al*. [54] combine spatial and temporal features from video sequences for human activity recognition and Karpathy *et al*. [19] combine features from inputs separated by 15 frames in a video for classification. Hei Ng *et al*. [32] proposed several convolution and pooling operations to combine features for video classification while Sistu *et al*. [51] used simple 1×1 bottleneck convolutions to combine features from consecutive frames for video segmentation.

In automotive or indoor robotic visual perception problems, simple concatenation techniques perform well but they fall short in some applications like video captioning [10, 33] or summarization [42] where long term dependencies are required. LSTMs in such cases offer a better alternative [59, 45]. Convolution-LSTMs (Conv-LSTMs) [58, 53] and 3D convolutions [18] are other options. However, these options incur additional computational complexity and they are needed mainly for aggregation of features that are significant for long term dependencies.

2.3. Multi-Task Loss

With the growing popularity of MTL, it is worth considering the possibility of imbalances in training an MTL network. It is often observed that some tasks dominate others during the training phase [14]. This dominance can be attributed to variations in task heuristics like complexities, uncertainties, and magnitudes of losses etc. Therefore an appropriate loss or prioritization strategy for all tasks in an MTL is a necessity.

Early works in MTL [55, 22], use a weighted arith-

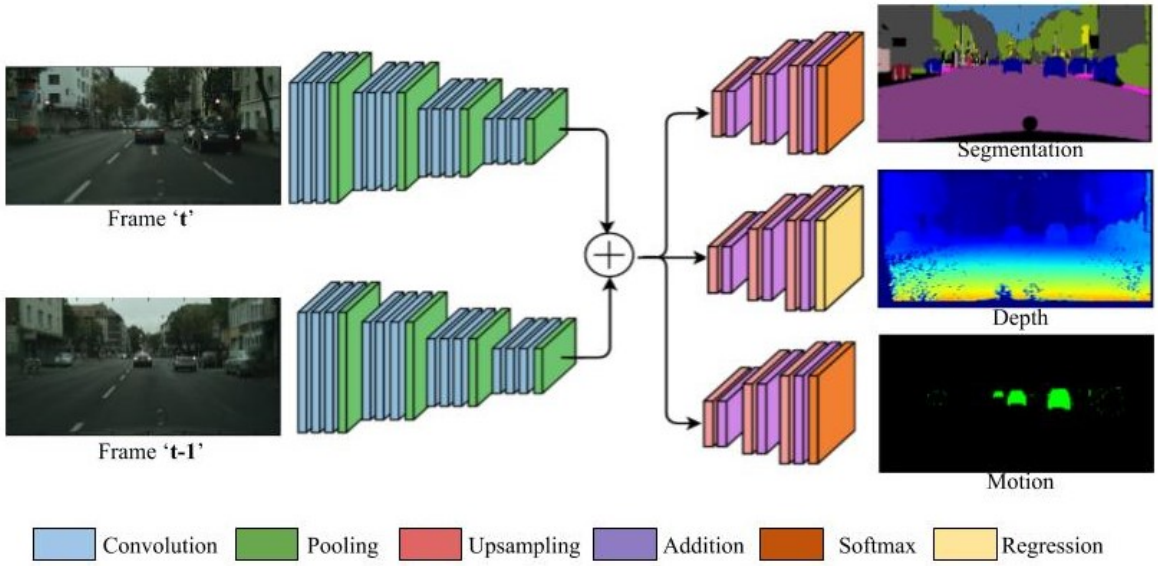


Figure 2: Illustration of the MultiNet++ network operating on consecutive frames of input video sequence. Consecutive frames are processed by a shared siamese-style encoder and extracted features are concatenated and processed by task specific segmentation, depth estimation and moving object detection decoders.

metic sum of individual task losses. Later, several works attempted to balance the task weights using certain task heuristics discussed earlier. Kendall *et al*. [20] proposed to use homoscedastic uncertainty of tasks to weigh them. This approach requires explicit modeling of uncertainty and more importantly, the task weights remain constant.

GradNorm [3] is another notable work in which Chen *et al*. proposes to normalize gradients from all tasks to a common scale during backpropagation. Lui *et al*. [26] proposed Dynamic Weight Average (DWA) which uses an average of task losses over time to weigh the task losses. Guo *et al*. [14] on the other hand proposed dynamic task prioritization where the changes in the difficulty of tasks adjust the task weights. This allows distributing focus on harder problems first and then on less challenging tasks. On another hand, Liu *et al*. devised a different strategy to use a reinforcement learning based approach to learn optimal task weights. However, this method isn't simple and it brings additional complexity to the training phase.

In contrast to modeling multi-task problem as a single objective problem, Sener and Koltun [44] proposed to model it as a multi-optimization problem. Zhang and Yeung [61] proposed a convex formulation for multi-task learning and Desideri [7] proposed a multiple-gradient descent algorithm. In summary, these strategies either involve an explicit definition of loss function using task heuristics or require complex optimization techniques. Therefore, a loss strategy with minimal design complexities will be well suited for multi-task learning to accommodate a virtually unlimited number of joint tasks.

3. Proposed Solution

We introduce our novel multi-task network MultiNet++, that is capable of processing multiple streams of input data. The proposed architecture is scalable and can be readily applied in any multi-task problem. In the following subsection, we discuss how we built our MultiNet++ network shown in Figure 2.

3.1. Multi-stream Multi-task Architecture

MultiNet++ is a simple multi-task network with the ability to process multiple streams of input data. It is built using three main components, 1) Encoders that feed multiple streams of input into the network, 2) Feature aggregation layers that concatenate the encoded feature vectors from multiple streams and 3) Task-specific decoders that operate on aggregated feature space to perform task-specific operations. In this paper, we use MultiNet++ for joint semantic segmentation, depth estimation and moving object detection (or simply motion) on video sequences. We share the encoder between two consecutive frames from a given video sequence as shown in Figure 2. This can significantly reduce the computational load as the encoders require a daunting number of parameters. These input frames can be selected sparsely or densely from a video sequence by observing its motion histogram. One can also choose to pass

keyframes as proposed by Kulhare *et al*. [23].

Our encoders are selected by removing fully connected layers from ResNet-50 [16]. Outputs from ReLU [15] activation at layers 23, 39 and 46 from ResNet-50 [16] encoder are extracted and sent to feature aggregation layers. These feature maps extracted from different streams of inputs are concatenated and sent to task-specific decoders as shown in Figure 1. Segmentation decoder is built using FCN8 [27] architecture that comprises of 3 upsampling layers and skip connections from aggregated feature maps as shown in Figure 2. The final layer consists of softmax [13] units to predict pixel-wise classification labels. Similarly, we construct a motion decoder by changing the number of output classes in softmax units. Depth decoder is built by replacing softmax with regression units.

3.2. Geometric Loss Strategy

We discussed the importance of a loss strategy that requires minimal effort during design phase in Section 2.3. The commonly used loss combination function is arithmetic mean and it suffers from differences in the scale of the individual losses. This is partially alleviated by weighted average of the losses but it is difficult to tune manually. We were motivated to explore geometric loss combination which is invariant to the scale of the individual losses. Thus we express the total loss of a multi-task learning problem as geometric mean of individual task losses. We refer to this as Geometric Loss Strategy (GLS). For an n-task problem with task losses '$\mathcal{L}_1$','$\mathcal{L}_2$' ... '$\mathcal{L}_n$', we express total loss as:

$$\mathcal{L}_{Total} = \prod_{i=1}^{n} \sqrt[n]{\mathcal{L}_i} \tag{1}$$

For example, in a 3-task problem with losses '$\mathcal{L}_1$','$\mathcal{L}_2$' and '$\mathcal{L}_3$', we express total loss:

$$\mathcal{L}_{Total} = \sqrt[3]{\mathcal{L}_1\mathcal{L}_2\mathcal{L}_3} \tag{2}$$

Equations 1 and 2 are quite popular in geometric programming. This loss function is differentiable and can be optimized using an optimizer like Stochastic Gradient Descent (SGD). In fact, this definition makes sure that all tasks are making progress. We adapt our loss function to focus or give more attention to certain tasks by introducing Focused Loss Strategy (FLS) where we multiply geometric mean of losses of focused tasks to existing loss function. In this case, we define loss function with focus on m $(m \leq n)$ important tasks as:

$$\mathcal{L}_{Total} = \prod_{i=1}^{n} \sqrt[n]{\mathcal{L}_i} \times \prod_{j=1}^{m} \sqrt[m]{\mathcal{L}_j} \tag{3}$$

Equation 3 provides an opportunity to focus on important tasks in a multi-task learning problem. Here we assume that the tasks are ordered in terms of priority so that first m tasks out of the total n tasks gets higher weightage.

Application of log function converts the product of losses to sum of log of individual losses and thus can be interpreted to be equivalent to normalizing individual losses and then adding them. However, it is computationally complex to make use of log function.

4. Experiments and Results

In this section, we discuss the datasets used for evaluating the efficacy of the proposed models. Later, we discuss in detail how we constructed the proposed models and provide a complexity analysis of each. We also discuss the optimization strategies used during the training phase. Finally, we provide the results obtained along with a discussion.

4.1. Datasets

KITTI [12], Cityscapes [6] and SYNTHIA [39] are popular automotive datasets. KITTI has annotations for several tasks including semantic segmentation, depth estimation, object detection, *etc*. However, these annotations were done separately for each task and the input is not always common across the tasks. KITTI Stereo 2015 [30, 29] dataset provides stereo images for depth estimation. A subset of these images is labeled for KITTI semantic segmentation [12]. This dataset consists of 200 train images and 200 test images. Cityscapes [6] dataset provides both segmentation and depth estimation annotations for $\approx$ 3500 images. Motion labels for these datasets are provided by Vertens *et al*. [56]. SYNTHIA [39] is a synthetic dataset that provides segmentation and depth annotations for raw video sequences simulated in different weather, light conditions and road types. KITTI [12] and Cityscapes [6] provide segmentation labels for 20 categories while SYNTHIA [39] dataset provides segmentation labels for 13 categories.

Annotations	KITTI[12]	Cityscapes[6]	SYNTHIA[39]
Segmentation	✓	✓	✓
Depth	✓	✓	✓
Motion	✓	✓	×
# Train	200	2,975	888
# Validation	200	500	787
# Type	Real	Real	Synthetic

Table 1: Summary of the automotive datasets used in our experiments.

In KITTI [12] and Cityscapes [6] datasets, images are sampled and annotated sparsely from raw videos. This poses a challenge to approaches that use temporal methods for segmentation or motion detection tasks in videos. In addition to KITTI [12] and Cityscapes [6] datasets, we use SEQS-02 (New York-like city) and SEQS-05 (New York-

Method	KITTI & Cityscapes					SYNTHIA			
	Encoder	Segmentation	Depth	Motion	Total	Encoder	Segmentation	Depth	Total
1-Task Segmentation, Depth or Motion									
1-Task	23.58M	0.18M	-	-	23.77M	23.58M	0.14M	-	23.68M
1-Task	23.58M	-	3.88K	-	23.59M	23.58M	-	3.87K	23.59M
1-Task	23.58M	-	-	8.33K	23.60M	-	-	-	-
2-Task Segmentation and Depth									
1-Frame	23.58M	0.18M	3.88K	-	23.77M	23.58M	95.34K	3.88K	23.69M
2-Frames	23.58M	0.26M	7.46K	-	23.86M	23.58M	0.14M	7.46K	23.74M
2-Task Segmentation and Motion									
1-Frame	23.58M	0.18M	-	8.33K	23.78M	-	-	-	-
2-Frames	23.58M	0.26M	-	15.50K	23.86M	-	-	-	-
3-Task Segmentation, Depth and Motion									
1-Frame	23.58M	0.18M	3.88K	8.33K	23.79M	-	-	-	-
2-Frames	23.58M	0.26M	7.46K	15.50K	23.87M	-	-	-	-

Table 2: Comparative study: Parameters needed to construct 1-task segmentation, depth and motion, 2-task segmentation and depth, 2-task segmentation and motion and 3-task segmentation, depth and motion models. We compare 2-task and 3-task models that operate on 1-frame and 2-frames.

like city) from SYNTHIA dataset for training and validation respectively in our experiments. These sequences provide segmentation and depth annotations for consecutive images in a video sequence. Thus they are more suitable for evaluating our multi-task model which operates on multiple streams of input data. Table 1 provides a summary of different properties of the 3 datasets discussed so far.

4.2. Model Analysis

We constructed several models to evaluate the benefits of the proposed MultiNet++. We build 3 single task baseline models for segmentation, depth and motion tasks using ResNet-50 [16] as an encoder and different task-specific decoders as discussed in Section 3.1. Segmentation decoder predicts pixel-wise labels from 20 different categories for input in KITTI [12] & Cityscapes [6] datasets, while the decoder predicts from 13 categories in SYNTHIA [39] dataset. Depth decoder outputs a 16-bit integer at every pixel location to predict depth and motion decoder predicts a binary classification label for every pixel to classify as moving or static object. These models process one frame of input data. We also constructed 2-task and 3-task models that operate on a single frame and 2 consecutive frames of an input video sequence. MultiNet++ refers to models that operate on 2 consecutive frames which are built using feature aggregation as discussed in Section 3.1. Table 2 provides details about number parameters required to construct different models.

Majority of computational load arises from ResNet-50 [16] encoder. Due to this property, 2-task and 3-task models required the almost same number of parameters as 1-task model. This is one of the main reasons why multi-task networks are computationally efficient and favor embedded deployment. We build our 2-frame models with relatively very little increase in complexity ($\approx$ 100K parameters) by reusing the encoder between 2-frames. In 2-frames model, the aggregated features are larger in size when compared to the 1-frame model. It resulted in an increase of parameters.

4.3. Optimization

We implemented our proposed models using Keras [5]. In all our experiments, we re-size the input images to 224×384. We used only 2-frames for feature aggregation because adding more frames would increase computational complexity with insignificant performance gains as demonstrated by Sistu *et al.* [51]. In our multi-task learning networks, we define the loss functions for each task separately and feed them to our geometric loss strategy (GLS) proposed in Section 2.3. For semantic segmentation and motion, we use pixel-wise cross-entropy loss for C classes averaged over a mini-batch with N samples as shown in Equation 4.

$$\mathcal{L}_{Seg} \text{ or } \mathcal{L}_{Motion} = -\sum_{j=1}^{N}\sum_{i=1}^{C} y_{i,j} log(p_{i,j}) \tag{4}$$

For depth estimation, we use Huber loss as defined in Equation 5 with δ =250.

$$\mathcal{L}_{Depth} = \begin{cases} \frac{1}{2}\left[y-\hat{y}\right]^2 & : |y-\hat{y}| \leq \delta \\ \delta\left(|y-\hat{y}| - \delta/2\right) & : otherwise \end{cases} \tag{5}$$

The total loss $\mathcal{L}_{Total}$ is defined as:

$$\mathcal{L}_{Total} = \sqrt[3]{\mathcal{L}_{Seg}\mathcal{L}_{Depth}\mathcal{L}_{Motion}} \tag{6}$$

Method	KITTI			Cityscapes			SYNTHIA	
	Segmentation	Depth	Motion	Segmentation	Depth	Motion	Segmentation	Depth
1-Task Segmentation, Depth or Motion								
1-Task	81.74%	-	-	78.95%	-	-	84.08%	-
1-Task	-	75.91%	-	-	60.13%	-	-	73.19%
1-Task	-	-	**98.49**%	-	-	**98.72**%	-	-
2-Task Segmentation and Depth								
Equal weights	74.30%	74.47%	-	73.76%	59.38%	-	63.45%	71.84%
GLS (ours)	81.50%	74.92%	-	79.14%	60.15%	-	86.87%	73.60%
MultiNet++	81.01%	73.95%	-	**83.07**%	60.15%	-	**88.15**%	**78.39**%
2-Task Segmentation and Motion								
Equal weights	80.14%	-	97.88%	78.46%	-	98.25%	-	-
GLS (ours)	81.52%	-	97.93%	77.63%	-	98.83%	-	-
MultiNet++	81.75%	-	98.15%	78.86%	-	98.65%	-	-
3-Task Segmentation, Depth and Motion								
Equal weights	77.14%	76.15%	97.83%	72.71%	60.97%	98.20%	-	-
GLS (ours)	**82.20**%	**76.54**%	97.92%	77.38%	61.56%	98.72%	-	-
MultiNet++	80.06%	73.94%	97.94%	82.36%	**62.74**%	98.21%	-	-

Table 3: Improvements in learning segmentation, depth estimation and motion detection as multiple tasks using equal weights, proposed geometric loss strategy (GLS) and 2 stream feature aggregation with GLS (MultiNet++) vs independent networks (1-Task) on KITTI, Cityscapes and SYNTHIA datasets.

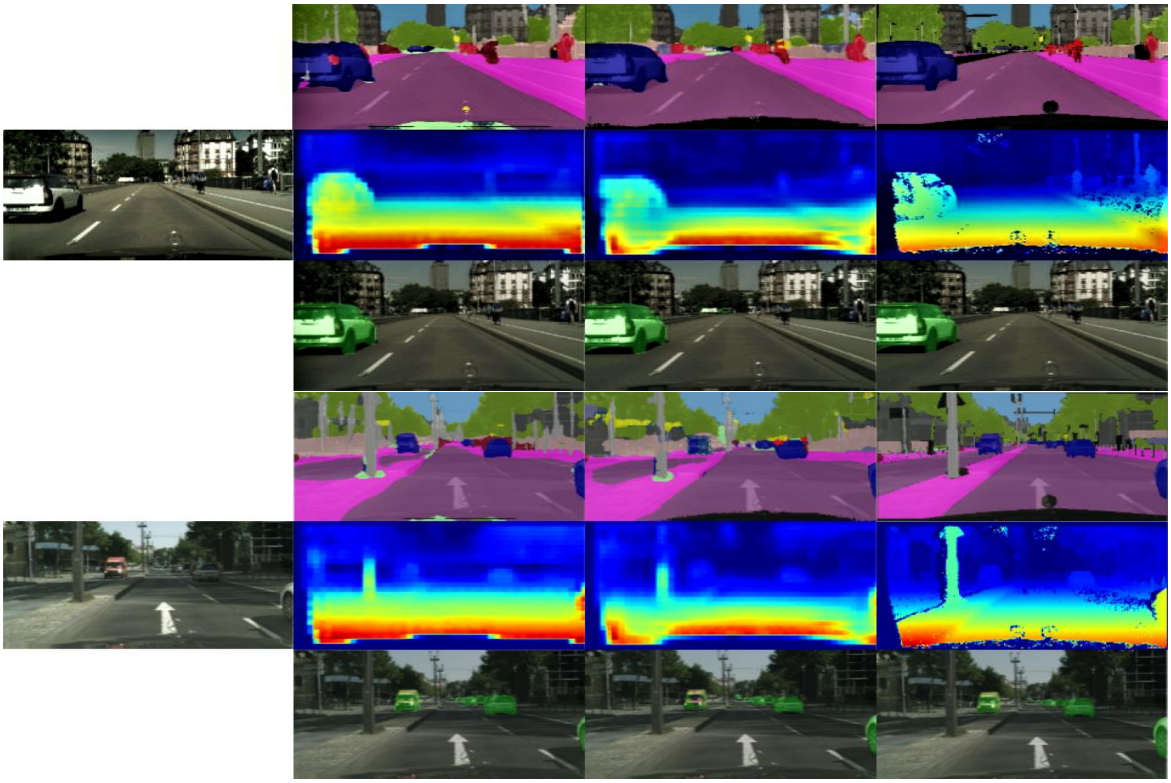
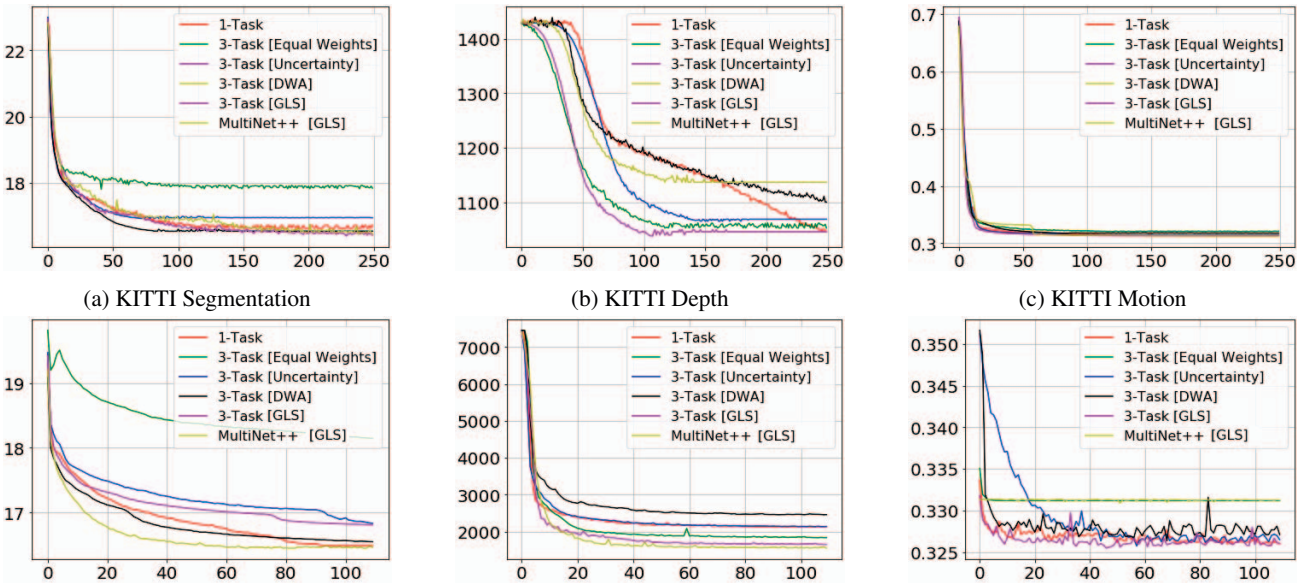


Figure 3: Left to Right: Input Image, Single Task Network outputs, MultiNet++ Output, Ground Truth. More qualitative results of MultiNet++ model can be accessed via this link https://youtu.be/E378PzLq7lQ.



(a) KITTI Segmentation (b) KITTI Depth (c) KITTI Motion

(d) Cityscapes Segmentation (e) Cityscapes Depth (f) Cityscapes Motion

Figure 4: Change of validation loss (X-axis) over several epochs (Y-axis) during training phase for 1-Task model vs 3-Task models for segmentation, depth and motion tasks on KITTI [12] and Cityscapes [6] datasets.

We optimize this loss function in our training phase using Adam optimizer [21]. Accuracy is used as an evaluation metric for segmentation and motion tasks while regression accuracy is used for depth estimation.

4.4. Results

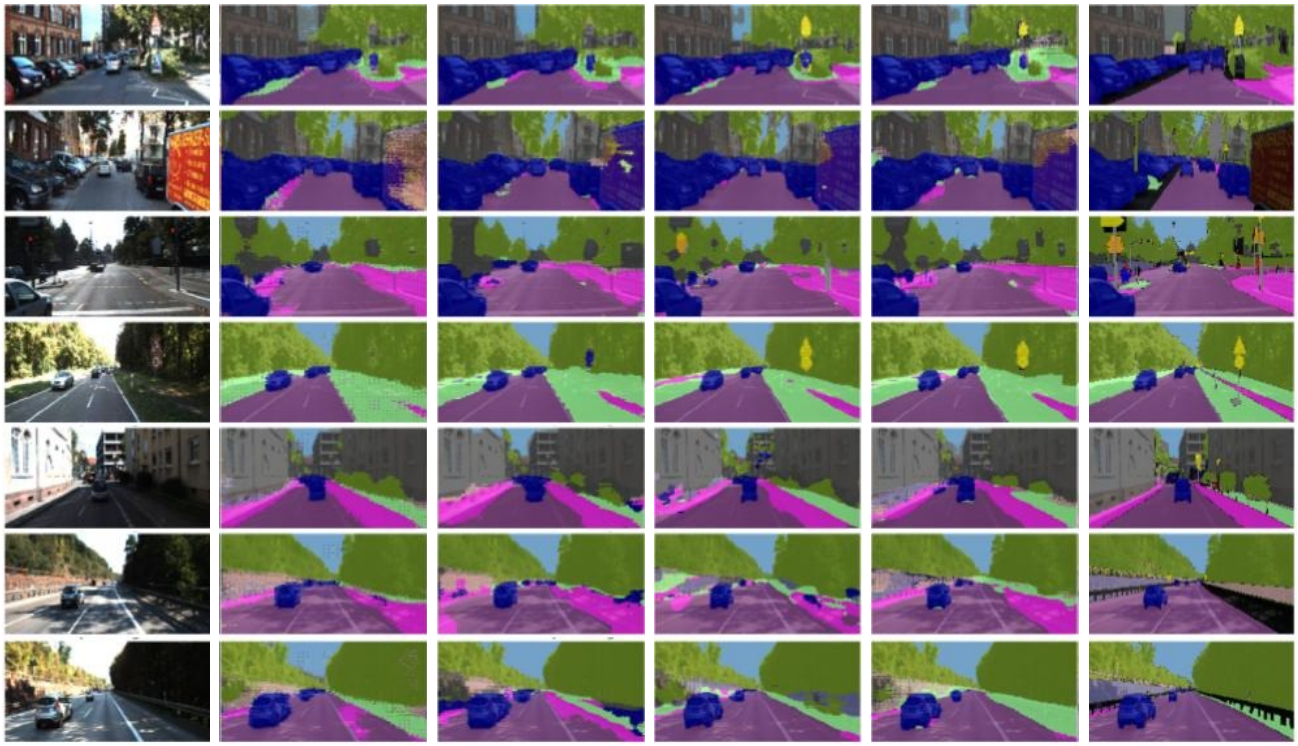
In Table 3, we compare the results of 2-task models and 3-task models using our geometric loss strategy (GLS) against naive equal task weight method. We also compare their performances with 1-task segmentation, depth and motion models. Our GLS method shows significant improvements in performance over equal weights method in both 2-task and 3-task models. In Table 4, we compare the results of 3-task models using our geometric loss strategy (GLS) against naive equal task weights, uncertainty weight method proposed by Kendal *et al*. [20] and Dynamic Weight Average (DWA) proposed by Liu *et al*. [26]. In Figure 4 (4a, 4b, 4c, 4d, 4e and 4f), we show how validation loss for these models change over time during training phase. Our models using GLS demonstrated faster convergence on all tasks. In 3-task models solving for segmentation, depth, and motion, depth is usually the most complex task. Figures 4b and 4e show that depth estimation on KITTI [12] and Cityscapes [6] requires longer convergence time compared to segmentation (Figures 4a and 4d) and motion tasks (Figures 4c and 4f). In these cases, our GLS method has shown faster convergence compared to uncertainty [20] and DWA [26] methods. While solving for multiple tasks, uncertainty [20] and DWA [26] weigh the tasks that converge quickly higher than

Method	Segmentation	Depth	Motion
KITTI			
1-Task	81.74%	75.91%	**98.49%**
Equal weights	77.14%	76.15%	97.83%
Uncertainty [20]	78.93%	75.73%	98.00%
DWA [26]	80.05%	74.48%	97.78%
GLS (ours)	**82.20%**	**76.54%**	97.92%
Cityscapes			
1-Task	**78.95%**	60.13%	**98.72%**
Equal weights	72.71%	60.97%	98.20%
Uncertainty [20]	77.32%	60.44%	98.63%
DWA [26]	78.05%	59.34%	98.45%
GLS (ours)	77.38%	**61.56%**	**98.72%**

Table 4: Comparative Study: Performance of 1-Task, equal weights, 3-task uncertainty [20], Dynamic Weight Average (DWA) [26] and proposed geometric loss strategy (GLS) on KITTI and Cityscapes datasets.

the others. This led to faster convergence in segmentation and motion tasks but late convergence in depth task. In such circumstances, the encoder parameters might be biased towards segmentation and motion tasks. This can result in imbalanced learning of depth task. Our GLS method expresses the total loss as the geometric mean of individual losses, so it doesn't prioritize one task higher than others. In this way, we achieve balanced training and improved performances compared to other techniques.

In Table 3, we also compare 2-task and 3-task mod-



(a) Input Images (b) 1-Task (c) Equal Weights (d) GLS (e) MultiNet++ (f) Ground Truth

Figure 5: Comparison of Semantic Segmentation results: 1-Task Segmentation vs 3-Task models on KITTI dataset.

els with our novel MultiNet++ which uses both feature aggregation (for 2-frame input) and GLS. In KITTI [12] dataset, input images are sparsely sampled from raw video sequences which hinder the performance gains of MultiNet++. In Cityscapes [6] dataset, MultiNet++ outperforms single task models by 4% and 3% for segmentation and depth tasks respectively as they provide images sampled closely compared to KITTI dataset. These improvements are much better in SYNTHIA [39] dataset (4% and 5% for segmentation and depth estimation tasks respectively) as they provide continuous frames of video sequences. We achieve similar performances for motion task compared to 1-task models.

We compare qualitative results of MultiNet++ with 1-task segmentation model on Cityscapes [6] dataset in Figure 3. The main difference between 1-task models and 3-task models is that the latter have learned representations from other tasks using a common encoder. Knowledge acquired through these representations helps 3-task model to identify semantic boundaries better compared to 1-task model. It is clearly evident that MultiNet++ model has improved performance. Our models detect traffic signs, lights and other near range objects better compared to other models on KITTI dataset [12] as shown in Figure 5.

5. Conclusion

We introduced an efficient way of constructing MultiNet++, a multi-task learning network that operates on multiple streams of input data. We demonstrated that our geometric loss strategy (GLS) is robust to different task heuristics like complexity, magnitude, *etc*. We achieved balanced training and improved performances for a multi-task learning network solving different tasks namely segmentation, depth estimation and motion on automotive datasets KITTI, Cityscapes, and SYNTHIA. Our GLS strategy is easy to implement and most importantly it allows for balanced learning of a large number of tasks in multi-task learning without requiring explicit loss modeling when compared to other multi-task learning loss strategies. In the future, we would like to explore the benefits of multi-task learning networks using our efficient feature aggregation and loss strategies for multi-modal data.

Acknowledgements

Authors would like to thank their employer for supporting fundamental research. Authors would also like to thank Dr. Aditya Viswanathan and Dr. Thibault Julliand for helpful discussions.

References

[1] H. Bilen and A. Vedaldi. Universal representations: The missing link between faces, text, planktons, and cat breeds. *arXiv preprint arXiv:1701.07275*, 2017. 2

[2] R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, Jul 1997. 1, 2

[3] Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *ICML*, 2018. 3

[4] S. Chennupati, G. Sistu., S. Yogamani., and S. Rawashdeh. Auxnet: Auxiliary tasks enhanced semantic segmentation for automated driving. In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP,*, pages 645–652. INSTICC, SciTePress, 2019. 1

[5] F. Chollet et al. Keras. https://keras.io, 2015. 5

[6] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 4, 5, 7, 8

[7] J.-A. Désidéri. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5-6):313–318, 2012. 3

[8] C. Devin, A. Gupta, T. Darrell, P. Abbeel, and S. Levine. Learning modular neural network policies for multi-task and multi-robot transfer. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2169–2176, May 2017. 1

[9] P. Dewangan, S. P. Teja, K. M. Krishna, A. Sarkar, and B. Ravindran. Digrad: Multi-task reinforcement learning with shared actions. *CoRR*, abs/1802.10463, 2018. 1

[10] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):677691, Apr 2017. 2

[11] D. Dong, H. Wu, W. He, D. Yu, and H. Wang. Multi-task learning for multiple language translation. In *ACL*, 2015. 1

[12] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 2, 4, 5, 7, 8

[13] I. Goodfellow, Y. Bengio, and A. Courville. Deep learning. *MIT Press*, pages 189–191, 2016. http://www.deeplearningbook.org. 4

[14] M. Guo, A. Haque, D.-A. Huang, S. Yeung, and L. Fei-Fei. Dynamic task prioritization for multitask learning. In *European Conference on Computer Vision*, pages 282–299. Springer, 2018. 2, 3

[15] R. H. Hahnloser and H. S. Seung. Permitted and forbidden sets in symmetric threshold-linear networks. In *Advances in Neural Information Processing Systems*, pages 217–223, 2001. 4

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016. 4, 5

[17] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997. 2

[18] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2013. 2

[19] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. 1, 2

[20] A. Kendall, Y. Gal, and R. Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3, 7

[21] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2014. 7

[22] I. Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5454–5463, July 2017. 1, 2

[23] S. Kulhare, S. Sah, S. Pillai, and R. Ptucha. Key frame extraction for salient activity recognition. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 835–840, Dec 2016. 4

[24] J. Lee and J. Nam. Multi-level and multi-scale feature aggregation using pretrained convolutional neural networks for music auto-tagging. *IEEE signal processing letters*, 24(8):1208–1212, 2017. 2

[25] S. Liu. *Exploration on Deep Drug Discovery: Representation and Learning*. PhD thesis, University of Wisconsin-Madison, 2018. 1

[26] S. Liu, E. Johns, and A. J. Davison. End-to-end multi-task learning with attention. *arXiv preprint arXiv:1803.10704*, 2018. 3, 7

[27] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 4

[28] L. Ma, J. Stückler, C. Kerl, and D. Cremers. Multi-view deep learning for consistent semantic mapping with rgb-d cameras. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 598–605. IEEE, 2017. 1

[29] M. Menze, C. Heipke, and A. Geiger. Joint 3d estimation of vehicles and scene flow. In *ISPRS Workshop on Image Sequence Analysis (ISA)*, 2015. 4

[30] M. Menze, C. Heipke, and A. Geiger. Object scene flow. *ISPRS Journal of Photogrammetry and Remote Sensing (JPRS)*, 2018. 4

[31] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert. Cross-stitch networks for multi-task learning. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016. 2

[32] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2015. 2

[33] R. M. Oruganti, S. Sah, S. Pillai, and R. Ptucha. Image description through fusion based recurrent multi-modal learning. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3613–3617. IEEE, 2016. 2

[34] B. Ramsundar, S. Kearnes, P. Riley, D. Webster, D. Konerding, and V. Pande. Massively multitask networks for drug discovery. 2015. *arXiv preprint arXiv:1502.02072*, 2015. 1

[35] A. Ranjan and M. J. Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4161–4170, 2017. 1

[36] R. Ranjan, V. M. Patel, and R. Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1):121135, Jan 2019. 2

[37] H. Rashed., S. Yogamani., A. El-Sallab., P. Křek, and M. El-Helw. Optical flow augmented semantic segmentation networks for automated driving. In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP,*, pages 165–172. INSTICC, SciTePress, 2019. 2

[38] S.-A. Rebuffi, H. Bilen, and A. Vedaldi. Efficient parametrization of multi-domain deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8119–8127, 2018. 2

[39] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3234–3243, 2016. 2, 4, 5, 8

[40] S. Ruder, J. Bingel, I. Augenstein, and A. Søgaard. Learning what to share between loosely related tasks. *arXiv preprint arXiv:1705.08142*, 2017. 2

[41] S. Sah. *Multi-Modal Deep Learning to Understand Vision and Language*. PhD thesis, Rochester Institute of Technology., 2018. 1

[42] S. Sah, S. Kulhare, A. Gray, S. Venugopalan, E. Prud'Hommeaux, and R. Ptucha. Semantic text summarization of long videos. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 989–997. IEEE, 2017. 2

[43] V. Sanh, T. Wolf, and S. Ruder. A hierarchical multi-task approach for learning embeddings from semantic tasks, 2018. 1

[44] O. Sener and V. Koltun. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems*, pages 525–536, 2018. 3

[45] S. Sharma, R. Kiros, and R. Salakhutdinov. Action recognition using visual attention. In *Proceedings of the IEEE international conference on computer vision*, pages 4507–4515, 2015. 2

[46] M. Siam, S. Elkerdawy, M. Jagersand, and S. Yogamani. Deep semantic segmentation for automated driving: Taxonomy, roadmap and challenges. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8. IEEE, 2017. 1

[47] M. Siam, H. Mahgoub, M. Zahran, S. Yogamani, M. Jagersand, and A. El-Sallab. Modnet: Motion and appearance based moving object detection network for autonomous driving. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2859–2864. IEEE, 2018. 1

[48] M. Siam, S. Valipour, M. Jägersand, N. Ray, and S. Yogamani. Convolutional gated recurrent networks for video semantic segmentation in automated driving. *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–7, 2017. 1

[49] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014. 1

[50] O. Siohan and D. Rybach. Multitask learning and system combination for automatic speech recognition. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 589–595, Dec 2015. 1

[51] G. Sistu., S. Chennupati, and S. Yogamani. Multi-stream cnn based video semantic segmentation for automated driving. In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP,*, pages 173–180. INSTICC, SciTePress, 2019. 1, 2, 5

[52] G. Sistu, I. Leang, S. Chennupati, S. Milz, S. Yogamani, and S. Rawashdeh. Neurall: Towards a unified model for visual perception in automated driving. *arXiv preprint arXiv:1902.03589*, 2019. 2

[53] H. Song, W. Wang, S. Zhao, J. Shen, and K.-M. Lam. Pyramid dilated deeper convlstm for video salient object detection. In *The European Conference on Computer Vision (ECCV)*, September 2018. 2

[54] L. Sun, K. Jia, D.-Y. Yeung, and B. E. Shi. Human action recognition using factorized spatio-temporal convolutional networks. *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015. 2

[55] M. Teichmann, M. Weber, M. Zllner, R. Cipolla, and R. Urtasun. Multinet: Real-time joint semantic reasoning for autonomous driving. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1013–1020, June 2018. 1, 2

[56] J. Vertens, A. Valada, and W. Burgard. Smsnet: Semantic motion segmentation using deep convolutional neural networks. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 582–589, Sep. 2017. 1, 4

[57] Z. Wu, C. Valentini-Botinhao, O. Watts, and S. King. Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4460–4464, April 2015. 1

[58] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015. 2

[59] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing videos by exploiting tempo-

ral structure. In *Advances in Neural Information Processing Systems*, 2015. 2

[60] F. Yu, D. Wang, E. Shelhamer, and T. Darrell. Deep layer aggregation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[61] Y. Zhang and D.-Y. Yeung. A convex formulation for learning task relationships in multi-task learning. In *UAI*, 2010. 3

[62] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei. Flow-guided feature aggregation for video object detection. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2