Adversarial Multi-Task Learning Framework with GANs for Joint Semantic Segmentation and Depth Estimation with Perceptual Loss Integration

Anonymous CVPR submission

Paper ID *****

Abstract

This paper describes a new adversarial MTL framework to solve semantic segmentation and depth estimation problems jointly. Building on the Cityscapes dataset, the solution utilizes a small MobileNetV3-Small encoder, shared and task-based generators, and discriminators in a GAN architecture. Combining loss functions such as Cross-Entropy, Dice, Scale-Invariant, Perceptual and Adversarial losses gives robust, accurate and perceptually consistent results. The two-level GAN architecture combines task-level objectives and joint feature learning to understand urban scenes.

1. Introduction

Multi-task learning (MTL) has emerged as a revolutionary machine learning solution enabling a single model to perform more than one task by using common representations. In contrast to task-specific models, MTL achieves more efficiency and generalization by aggregating associated tasks in a single model. For instance, in computer vision applications, tasks such as object detection, semantic segmentation and depth estimation, have similar patterns that MTL taps into to obtain a better performance [9]. Since MTL allows the exchange of knowledge between tasks, this feature makes MTL ideally suited for real-time scenarios where speed of computation and inference are critical.

Applications requiring real-time decision-making, like autonomous cars and robotics, depend on fast, accurate decisions. MTL meets these needs by aggregating computational resources in one framework, which helps minimize latency and allows faster inference without performance loss [4]. For example, in autonomous driving, a single MTL model can find objects, calculate their distance and make sense of lane markings, making it faster to decide. This eliminates the need for multiple models and allows real-time processing, even in limited resource environments such as vehicle-embedded systems [1].

MTL is now being adopted by the automotive industry as a foundation for developing autonomous vehicle technologies. Incorporating activities such as semantic segmentation, depth prediction, and object recognition, MTL boosts the autonomy of self-driving cars. For instance, Tesla's Autopilot and other autonomous driving systems use deep learning models that are also programmed to identify roadside objects, calculate their distances, and estimate their paths. Performing these individually would require many different models with similar functionality, increasing computation costs and latency. MTL combines these tasks into one network for a smooth and coordinated operation [8]. Similarly, Waymo's self-driving cars employ MTL to perform collaborative tasks such as path planning, pedestrian detection, and lane following, providing intelligent and safe navigation in dense cities.

The challenges of semantic segmentation and depth estimation are essential to the use of MTL in autonomous vehicles. Semantic segmentation means a label being placed on each pixel in an image, which will help the system identify the world around it on a micro-level. For example, it detects traffic signals, vehicles, pedestrians and obstructions and offers contextual information needed to navigate [6]. Depth estimation goes even further, calculating the object distance from the camera, which is crucial to spatial perception and collision avoidance [2]. Together, these tasks form the basis of scene interpretation, which allows the vehicle to make accurate sense of its surroundings.

Imagine driving through a crowded urban intersection with an autonomous car, simultaneously identifying people walking across the street (semantic segmentation), estimating the distance of other vehicles (depth estimation), and distinguishing traffic signals (object detection). MTL lets a single model receive input from sensors and produce outputs for all these tasks in real time, thereby greatly enhancing computational effectiveness and decision speed [7]. This capability is necessary in time-sensitive applications, where delaying processes might lead to accidents or reduced efficiency.

MTL offers a powerful method to tackle several tasks si-

multaneously but can be extended with complementary approaches such as Generative Adversarial Networks (GANs). GANs deliver a transformative potential by creating high-quality synthetic data and domain adaptation, which solves the most pressing real-time application challenges of data scarcity and domain variation [11]. Besides creating synthetic data, GANs can also be used to create shared features for MTL. With their adversarial training mechanism, GANs can identify and refine shared data structures between tasks. For example, in autonomous driving, semantic segmentation and depth estimation both use common properties, such as the shape of objects and spatial connectivity. A GAN can generate synthetic backbone features, enabling the MTL model to utilize these shared representations effectively across multiple tasks. This ability minimizes redundancy and facilitates generalization so that the MTL model can still work well even under dark conditions.

Apart from facilitating shared feature learning, GANs facilitate the conversion between real-world and synthetic data distribution, an essential aspect of domain adaptation. GANs mimic the different environments of a city, whether it's a rainy day, under different lights, or jammed with traffic, to extend the training process and make MTL models responsive to urban realities [10]. A GAN-constructed dataset might contain edge-cases, for example, unusual road shapes or sudden pedestrian arrivals, ensuring that the resulting common features from the MTL model are strong and contextually diverse. In addition, multi-task discriminators and generators in GANs helps the model cope with more complex tasks. Multi-task discriminators can compare outputs of more than one task at a time, leading to improved generalization and discrimination, and multi-task generators yield outputs optimized for more than one goal. This partnership enhances diversity and representation learning in difficult, real-world situations [5], [12]. This enhanced design enables GANs to interact harmoniously with MTL structures, leveraging task-specific details and ensuring task-coherent learning.

In this paper, we are interested in using MTL for semantic segmentation and depth estimation with Cityscapes dataset, a standard for urban scene analysis dataset. We also use GAN-created data to avoid data bottlenecks and support shared feature learning. This combination shows how GANs complement MTL, both by enriching training data and by creating robust shared representations. These findings confirm the mutually beneficial utility of merging MTL and GANs to create stronger, more capable systems, especially in the autonomous vehicle field where safety and accuracy are key.

2. Research Pre-requisite

In this section we discuss about the Dataset and loss functions used in this study.

2.1. Dataset

Cityscapes is a set of street-level urban scenes for use in computer vision research applications such as semantic segmentation, depth estimation and disparity map generation [3]. It offers high-resolution imagery and ground truth annotations, which can be extremely helpful in learning the complexities of urban environments.

To estimate depth, the dataset's leftImg8bit and rightImg8bit subsets are used to generate stereo image pairs that are used to calculate disparity maps using CreStereo models. The gtFine set, which contains pixel-level annotations, allows semantic segmentation to get accurate urban scene parsing and object categorization.

CreStereo is a high-end deep learning model of stereo depth estimation, which can handle stereo image pairs, two identical images from slightly different left and right camera angles. The model predicts a disparity map, which is a 2D image in which pixel values represent the horizontal displacement between similar points in left and right images. The disparity values are inversely related to the depth of the objects in the scene. Combine the disparity map with camera settings, like focal length and baseline distance, and CreStereo will calculate a true depth map, which gives the distance objects are from the camera. The leftImg8bit and rightImg8bit subdirectories on the Cityscapes dataset can be used as a stereo image pair that the CreStereo model uses to compute disparity and depth maps. These outputs are crucial for applications such as training monocular depth estimation, scene recognition and autonomous driving. In the computation of disparity maps on the Cityscapes dataset, a two-step coarse-to-fine refinement is performed. Firstly, stereo pictures are scaled down to a lower resolution with bilinear interpolation to save on computational cost while keeping spatial connections intact. These resized pictures are run through the CreStereo model to create a rough, raw disparity map. The second step extracts the original resolution images and fine-tunes the coarse disparity map to produce a high-resolution, accurate disparity map. The final disparity map is a 2D array that captures the pixel-wise horizontal displacement between stereo pictures and is used to estimate depth.

Making depth maps from disparity images translates pixel-wise horizontal displacement values into physical distances of the objects from the camera. When the disparity maps are created from stereo images, they are first preprocessed for noise, invalid regions, and inconsistencies. Methods like clipping irrelevant regions, scaling to common dimensions, and inpainting for missing values provide clean inputs. The noise is then filtered out with smoothing filters such as median blur to give the disparity map an improved quality. Upon preprocessing, disparity numbers are rescaled and normalized to a physical scale. Depth is calculated based on Equation 1, where the baseline is the dis-

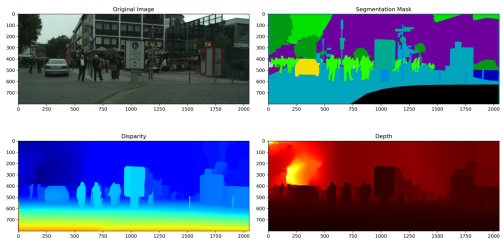

Figure 1. Example of the data used for our multi-task learning framework. Top-left: Original Image. Top-right: Segmentation Mask. Bottom-left: Disparity Map from CresStereo Model. Bottom-right: Depth Map generated from Disparity Maps.

tance between the stereo cameras, and the focal length is the optical configuration of the cameras. For the cityscapes dataset configuration, a baseline of 0.209313 meters and a focal length of 2262.52 pixels are considerd. Depth values are usually trimmed to the maximum depth considered to keep outputs as realistic as possible. The resultant depth maps are accurate and repeatable, which is ideal for 3D reconstruction, generative model training, autonomous navigation and robotics.

$$\text{Depth} = \frac{\text{Baseline} \cdot \text{Focal Length}}{\text{Disparity} + \epsilon} \tag{1}$$

2.2. Loss Function and there application in MTL

In multitask learning architectures, loss functions help to balance goals including semantic segmentation, depth estimation and adversarial training. This setup often includes task-dependent and joint losses to get the best performance out of each task and to get the advantage of sharing representations.

2.2.1 Cross-Entropy Loss

This loss is central to semantic segmentation problems, penalizing incorrect class predictions at the pixel level. It directs the network to assign high probabilities ($\hat{y}_{i,c}$) to the correct class for each pixel ($y_{i,c}$), ensuring accurate classification of image regions.

$$\mathcal{L}_{CE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(\hat{y}_{i,c}) \tag{2}$$

2.2.2 Dice Loss

Dice Loss minimizes the overlap error between the predicted segmentation mask ($\hat{y}$) and the ground truth (y). It is particularly effective on unbalanced datasets, optimizing for maximum similarity between predictions and targets by focusing on overlap rather than individual pixel classifications. The Dice Loss is computed as:

$$\mathcal{L}_{Dice} = 1 - \frac{2 \cdot \text{Intersection} + \epsilon}{\text{Union} + \epsilon} \tag{3}$$

Here, Intersection refers to the sum of element-wise multiplications of the prediction ($\hat{y}$) and ground truth (y), while Union is the sum of their individual values. ϵ is a small constant added to avoid division by zero. Dice Loss is particularly suitable for unbalanced datasets as it maximizes the similarity between predictions and targets by emphasizing the overlap between the predicted and ground truth segmentation masks.

2.2.3 c) Scale-Invariant Depth Loss

This loss captures the difference between the predicted depth ($\hat{d}_i$) and ground truth depth (d_i), focusing on relative differences instead of absolute values. This scale-invariance makes it robust across scenes with varying depth levels, making it ideal for diverse datasets. The scale-invariant loss is computed as:

$$\mathcal{L}_{SI} = \frac{1}{N}\sum_{i=1}^{N}(d_i - \hat{d}_i)^2 - \frac{\lambda}{N^2}\left(\sum_{i=1}^{N}(d_i - \hat{d}_i)\right)^2 \tag{4}$$

Here, N is the total number of pixels, d_i is the ground truth depth for pixel i, and $\hat{d}_i$ is the predicted depth for pixel i. The first term measures the mean squared error, while the second term introduces a penalty based on the mean error across all pixels, scaled by a factor λ. This formulation ensures that the loss focuses on relative differences, making it durable across a variety of depth levels and perfect for handling diverse scenes and datasets.

2.2.4 Inverse Huber Loss

Huber Loss transitions between quadratic and linear penalties, making it robust against large depth errors while still penalizing slight deviations. It is defined as:

$$\mathcal{L}_{Huber} = \begin{cases} 0.5 \cdot (|d - \hat{d}|)^2 & \text{if } |d - \hat{d}| \leq \delta, \\ \delta \cdot (|d - \hat{d}| - 0.5 \cdot \delta) & \text{if } |d - \hat{d}| > \delta \end{cases} \tag{5}$$

Here, d represents the ground truth depth, $\hat{d}$ is the predicted depth, and δ is the threshold at which the loss transitions from quadratic to linear. The quadratic penalty is applied for smaller errors, promoting smooth optimization, while the linear penalty is applied for larger errors, ensuring robustness against outliers.

2.2.5 Percepetual loss

Perceptual loss aligns predicted outputs ($\hat{y}$) with high-level feature maps ($\phi_l(\cdot)$) of the ground truth (y), extracted from a pretrained model, such as VGG16. This encourages semantic consistency for segmentation and depth tasks by focusing on structural and perceptual features rather than raw pixel differences. It is defined as:

$$\mathcal{L}_{Perceptual} = \sum_{l \in \text{Layers}} \|\phi_l(\hat{y}) - \phi_l(y)\|^2 \quad (6)$$

Here, $\phi_l(\cdot)$ represents the feature maps extracted from layer l of the pretrained model. By emphasizing high-level features, this loss promotes alignment in the semantic and structural aspects of the predictions, making it especially effective for tasks requiring perceptual fidelity.

3. Proposed Multi-Task Learning Framework

This work presents a multi-task learning framework leveraging the Cityscapes dataset to perform simultaneous semantic segmentation and depth estimation. The Cityscapes dataset contains high-resolution urban street images, including RGB images and their corresponding semantic segmentation labels. Ground truth depth maps have been generated using Crestereo algorithms. Preprocessing involves scaling images and labels to a quarter of their original resolution for computational efficiency, applying data augmentation techniques such as cropping, flipping, rotation, and color jittering for generalization, and normalizing RGB images using ImageNet statistics. Depth values are scaled and clipped logarithmically to ensure numerical stability.

The architecture is built around a pre-trained MobileNetV3-Small encoder, which extracts top-down features at different spatial scales. The encoder incorporates skip connections for preserving fine details. A shared generator processes these features through Conditional Refinement Pooling (CRP) blocks and convolutional layers, aggregating and refining task-independent feature maps for both tasks.

As illustrated in Figure 2, the proposed framework uses a MobileNetV3-Small backbone to extract features, which are further processed by a shared generator. Task-specific generators and discriminators refine these features for semantic segmentation and depth estimation, while a multi-task discriminator ensures consistent and realistic outputs.

In this multitask learning model, the integration of multiple loss functions ensures that each task contributes meaningfully to the overall performance. For semantic segmentation, Cross-Entropy Loss and Dice Loss are employed to enhance pixel-wise accuracy and mask quality. Perceptual Loss aligns predictions with high-level features, preserving semantic consistency and structural details. For monocular depth estimation, Scale-Invariant Loss, Inverse Huber Loss, and Smoothness Loss work together to improve depth predictions by ensuring accurate estimates, smooth transitions, and resistance to noise. Additionally, Adversarial Loss, implemented via GAN-based feedback, enforces realism in the outputs for both segmentation and depth estimation, enabling better generalization and reproducibility.

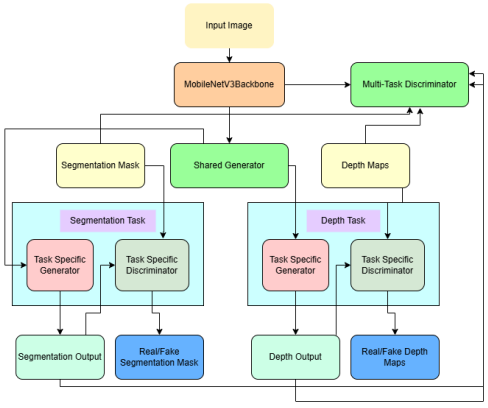

Figure 2. Overview of the proposed multi-task learning framework. The architecture consists of a MobileNetV3-Small backbone for feature extraction, a shared generator for refining task-independent features, and task-specific generators and discriminators for semantic segmentation and depth estimation.

By harmonizing these losses, the proposed framework optimizes overall performance while addressing the specific constraints of each task. This approach is particularly efficient in scenarios that require the simultaneous computation of semantic structure and depth information, leveraging common representations to balance competing goals and enhance model effectiveness.

4. Results and Analysis

The proposed multi-task learning framework was evaluated on the Cityscapes dataset, focusing on simultaneous semantic segmentation and depth estimation. The results demonstrate the model's ability to learn and generalize across both tasks effectively.

Figure 3 shows qualitative comparisons between ground truth (GT) and the predictions generated by the model. The results include RGB images, ground truth segmentation, ground truth depth, generated segmentation masks, and predicted depth maps. The generated outputs closely resemble the ground truth, highlighting the model's accuracy in capturing fine details in segmentation and smooth transitions in depth predictions.

Additionally, a GIF illustrating the evolution of segmentation and depth predictions across training epochs is included, showing gradual improvement in output quality as the training progresses. This dynamic visualization highlights the refinement of the model's predictions over time.

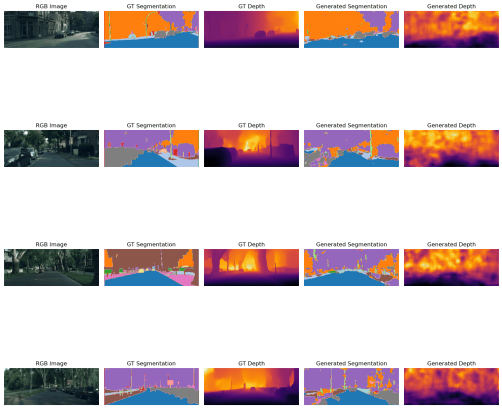

Figure 3. Qualitative results showing RGB images, ground truth segmentation and depth maps, and corresponding predictions from the model. The generated outputs demonstrate high consistency with the ground truth.

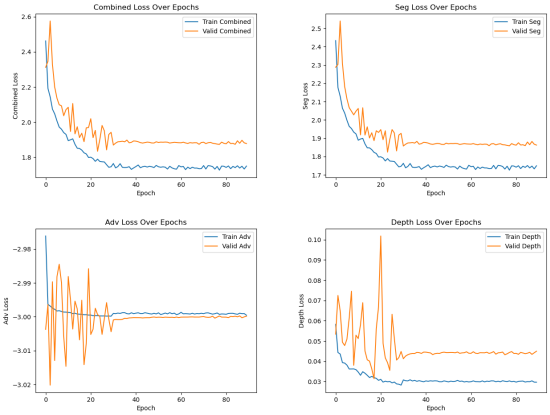

Figure 4. Training and validation loss curves over epochs for the proposed multi-task learning framework. (Top-left) Combined Loss, (Top-right) Segmentation Loss, (Bottom-left) Adversarial Loss, and (Bottom-right) Depth Loss. These plots illustrate stable convergence and generalization for both tasks.

The model's performance is analyzed through loss convergence curves, as shown in Figure 4. The plots include (Top-left) combined loss, (Top-right) segmentation loss, (Bottom-left) adversarial loss, and (Bottom-right) depth loss for both training and validation sets. The results indicate stable training dynamics, with all losses decreasing over time and achieving convergence. The adversarial loss contributes to the realism of the generated outputs, while the segmentation and depth losses highlight the task-specific optimization achieved during training.

4.1. Insights from Multi-Task Discriminator

The multi-task discriminator combines the output channels from all tasks to enforce consistency and realism in the generated outputs. This fusion mechanism contributes to improved segmentation mask quality and more accurate depth maps by harmonizing competing goals within the shared learning framework.

The combination of loss functions, including Cross-Entropy Loss, Dice Loss, Perceptual Loss, Scale-Invariant Loss, Inverse Huber Loss, Smoothness Loss, and Adversarial Loss, ensures the model optimally balances semantic segmentation and depth estimation tasks. The training framework demonstrates robustness and reproducibility, evidenced by consistent qualitative and quantitative performance.

5. Approach

This paper investigates an adversarial training approach for MTL that integrates semantic segmentation and depth estimation tasks. The model takes advantage of task-specific and collective loss functions, including adversarial losses, to maximize task performance on the individual task and to improve generalization across tasks. As you can see in Table 1, the models presented here are quite different when it comes to loss choices for generators and discriminators.

5.1. Model Configurations

5.1.1 Model 1 (test8)

Model 1 uses **Least Squares GAN (LSGAN)** loss for both segmentation and depth task discriminators. For the segmentation problem, generator losses include cross-entropy loss, dice loss, and perpetual segmentation loss. Scale-invariant depth Loss, Inverse Huber Loss, Depth Smoothness Loss, and Perpetual Depth Loss are the layers incorporated into the model to estimate depth. The shared generator loss equals the sum of task losses, and the multi-task discriminator uses the LSGAN loss. It is the default model.

5.1.2 Model 2 (test10)

For task-specific discriminators for segmentation and depth, Model 2 provides **Hinge Loss**. Segmentation generator losses do not change, such as Cross-Entropy Loss, Dice Loss, and Perpetual Segmentation Loss. Depth estimation losses are Scale-Invariant Depth Loss, Inverse Huber Loss, and Perpetual Depth Loss. Additionally, the shared generator loss combines segmentation and depth losses. The multi-task discriminator also exploits Hinge Loss, which targets adversarial consistency between tasks.

5.1.3 Model 3 (test11) - Best Model

Model 3, the most effective model, uses the **Wasserstein GAN with Gradient Penalty (WGAN-GP)** loss in the multi-task discriminator and segmentation and depth task discriminators and **Hinge loss** for task-specific generator and discriminator. Cross-Entropy Loss, Dice Loss and Perpetual Segmentation Loss are the Generator Losses for the segmentation function. In order to calculate the depth, the model employs Scale-Invariant Depth Loss, Inverse Huber Loss, and Perpetual Depth Loss. The aggregated task-specific generator loss maximizes the generator's efficiency in sifting through shared representations. This setup provides the highest stability and generality in both tasks.

5.1.4 Model 4 (test12)

Model 4 uses **WGAN-GP** loss in all discriminators, as does Model 3. But the shared generator loss and multi-task discriminator loss also fall within the scope of the WGAN-GP scheme. Generator losses were the same as in the earlier models, but this arrangement illustrates how adversarial training can balance task-specific and collective learning goals.

5.2. Loss Functions

The framework incorporates a combination of task-specific and adversarial losses:

- **Segmentation Losses**: Cross-Entropy Loss, Dice Loss, and Perpetual Segmentation Loss.
- **Depth Estimation Losses**: Scale-Invariant Depth Loss, Inverse Huber Loss, Depth Smoothness Loss, and Perceptual Depth Loss.
- **Adversarial Losses**: LSGAN, Hinge Loss, and WGAN-GP, applied to both task-specific and multi-task discriminators.

Model Name	Segmentation Task Specific Generator Loss (SegLoss)	Segmentation Task-Specific Discriminator	Depth Task Specific Generator (DepthLoss)	Depth Task Specific Discriminator	Shared Generator Loss	Multi-Task Discriminator loss
Model 1 (test8)	Cross Entropy Loss Dice Loss Perpetual Segmentation Loss	Least Square (LSGAN loss)	Scale Invariant Depth Loss Inverse Huber Loss Depth Smothness Loss Perpetual Depth Loss	Least Square (LSGAN loss)	SegLoss DepthLoss	Least Square (LSGAN loss)
Model 2 (test10)	Cross Entropy Loss Dice Loss Perpetual Segmentation Loss	Hinge loss	Scale Invariant Depth Loss Inverse Huber Loss Perpetual Depth Loss	Hinge Loss	SegLoss DepthLoss Combined generator loss	Hinge Loss
Model 3 (test11)	Cross Entropy Loss Dice Loss Perpetual Segmentation Loss	Hinge loss	Scale Invariant Depth Loss Inverse Huber Loss Perpetual Depth Loss	Hinge Loss	SegLoss DepthLoss Combined generator loss	WGAN-GP (Loss)
Model 4 (test12)	Cross Entropy Loss Dice Loss Perpetual Segmentation Loss	WGAN-GP (Loss)	Scale Invariant Depth Loss Inverse Huber Loss Perpetual Depth Loss	WGAN-GP (Loss)	SegLoss DepthLoss Combined generator loss	WGAN-GP (Loss)

Table 1. Loss configurations for all models.

5.3. Summary of Results

As shown in Table 1, Model 3 exhibits the best trade-off between segmentation and depth estimation performance. The integration of WGAN-GP loss ensures stability during adversarial training, while task-specific losses optimize individual task outputs. This demonstrates the effectiveness of combining adversarial training with task-specific loss functions in a multi-task learning framework. Following 5 and 6 displays the results of all are models in one frame.

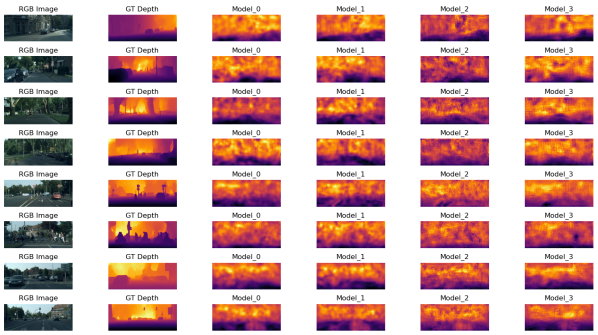

Figure 5. Depth estimation visualization for different models compared to the ground truth (GT).

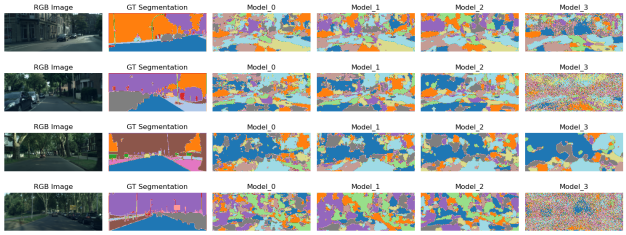

Figure 6. Semantic segmentation visualization for different models compared to the ground truth (GT).

6. Conclusion

In this paper, we presented a multi-task learning framework that integrates semantic segmentation and depth estimation using the Cityscapes dataset. By leveraging a pre-trained MobileNetV3-Small encoder, a shared generator, and task-specific components, the proposed architecture effectively balances task-specific goals and shared feature learning. The inclusion of various loss functions, including Cross-Entropy Loss, Dice Loss, Scale-Invariant Loss, Inverse Huber Loss, Perceptual Loss, and Adversarial Loss, ensures robust and accurate predictions while maintaining smoothness and realism in outputs.

The results demonstrate that the model can achieve stable convergence and generalize well across tasks. Qualitative and quantitative analyses illustrate the model's abil-

ity to produce high-quality segmentation masks and depth maps, closely aligning with ground truth. The use of adversarial feedback further enhances the outputs' realism and generalization.

This work highlights the potential of combining Multi-Task Learning (MTL) and Generative Adversarial Networks (GANs) to address challenges in real-time applications, such as autonomous driving. The integration of GANs supports shared feature learning and enables the model to adapt to diverse urban scenarios, ensuring robustness and efficiency. Future work could explore extending this framework to additional tasks and datasets, as well as incorporating real-world testing scenarios to validate its practical applicability.

7. Acknowledgement

This paper was prepared as part of the class project for CPSC 8810 - Machine Learning-Based Image Synthesis. The authors would like to express their heartfelt gratitude to Dr. Siyu Huang for her invaluable guidance and support throughout the project. Additionally, the authors acknowledge the assistance of OpenAI ChatGPT in refining grammatical errors and improving the storyline of the paper. Part of the work, including conceptual development and technical implementation, was independently completed by the authors as part of the project requirements.

References

[1] Inci M Baytas, Ming Yan, Anil K Jain, and Jiayu Zhou. Asynchronous multi-task learning. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 11–20. IEEE, 2016. 1

[2] Rohitash Chandra, Abhishek Gupta, Yew-Soon Ong, and Chi-Keong Goh. Evolutionary multi-task learning for modular training of feedforward neural networks. In *Neural Information Processing: 23rd International Conference, ICONIP 2016, Kyoto, Japan, October 16–21, 2016, Proceedings, Part II 23*, pages 37–46. Springer, 2016. 1

[3] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 2

[4] Yi He, Baijun Wu, Di Wu, and Xindong Wu. On partial multi-task learning. In *ECAI 2020*, pages 1174–1181. IOS Press, 2020. 1

[5] Wei Li, Li Fan, Zhenyu Wang, Chao Ma, and Xiaohui Cui. Tackling mode collapse in multi-generator gans with orthogonal vectors. *Pattern Recognition*, 110:107646, 2021. 2

[6] Kaixiang Lin and Jiayu Zhou. Interactive multi-task relationship learning. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 241–250. IEEE, 2016. 1

[7] Soyeon Park, Jiho Lee, and Eunwoo Kim. Resource-efficient multi-task deep learning using a multi-path network. *IEEE Access*, 10:32889–32899, 2022. 1

[8] Hoang Phan, Lam Tran, Ngoc N Tran, Nhat Ho, Dinh Phung, and Trung Le. Improving multi-task learning via seeking task-based flat regions. *arXiv preprint arXiv:2211.13723*, 2022. 1

[9] Trevor Standley, Amir Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In *International conference on machine learning*, pages 9120–9132. PMLR, 2020. 1

[10] Sandra Treneska, Eftim Zdravevski, Ivan Miguel Pires, Petre Lameski, and Sonja Gievska. Gan-based image colorization for self-supervised visual feature learning. *Sensors*, 22(4): 1599, 2022. 2

[11] Jianjin Xu, Zhaoxiang Zhang, and Xiaolin Hu. Extracting semantic knowledge from gans with unsupervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8):9654–9668, 2023. 2

[12] Ceyuan Yang, Yujun Shen, Yinghao Xu, Deli Zhao, Bo Dai, and Bolei Zhou. Improving gans with a dynamic discriminator. *Advances in Neural Information Processing Systems*, 35:15093–15104, 2022. 2