CAPSTONE PROJECT

Image segmentation for self driving cars

Presented By:

RAJ GOYAL-NIT Kurukshetra-Electrical Engineering

OUTLINE

- **Problem Statement**
- **Proposed System/Solution**
- **System Development Approach** (Technology Used)
- **Algorithm & Deployment**
- **Result**
- **Project Link**
- **Conclusion**
- **Future Scope**
- **References**

Problem Statement

- The rapid development of self-driving car technology has highlighted the critical need for accurate and efficient image segmentation systems. Image segmentation enables autonomous vehicles to precisely identify and classify various elements in their environment, such as pedestrians, vehicles, road signs, and lane markings. However, existing segmentation algorithms often struggle with the complexity and variability of real-world driving scenarios, leading to potential safety risks.

Proposed Solution

- The proposed system aims to enhance the reliability and safety of self driving cars by improving their ability to understand and navigate complex urban environments.
- Dataset Features:
  - Gathered the dataset that has still images from the original videos and the semantic segmentation labels shown in the images alongside the original image.
  - This dataset has 2975 training images files and 500 validation image files. Each image file is 256x512 pixels, and each file is a composite with the original photo on the left half of the image, alongside the labeled image (output of semantic segmentation) on the right half.
- Data Preprocessing:
  - Since, we have no label folders in this dataset, so we will not be able to use U-net CNN architecture model , as it is a supervised learning model, it requires labels of the corresponding images.
  - In order to use U-net, we will generate synthetic labels for all the images.
- Machine Learning Algorithm:
  - Implement a machine learning algorithm, such as U-net CNN pretrained model to predict mask based on the input images.
  - Consider data augmentation and using dropout layers in the architecture to improve prediction accuracy.
- Deployment:
  - Develop a user-friendly interface or application that performs image segmentation to precisely identify and classify various elements in their environment.
  - Deploy the solution on a scalable and reliable platform.
- Evaluation:
  - Assess the model's performance using appropriate loss function as binary cross-entropy and metrics as accuracy.
  - Fine-tune the model based on feedback and continuous monitoring of prediction accuracy.

System Approach

The "System Approach" section outlines the overall strategy and methodology for developing and implementing the image segmentation system for self-driving cars.

- **System requirements : Windows 11, 8GB RAM, Ryzen5 Processor.**
- **Library required to build the model: Numpy, Pandas, Matplotlib, Scikit-Learn, Scikit-Image**
- **Framework Used: Tensorflow**
- **Model: U-net CNN architecture.**

Algorithm & Deployment

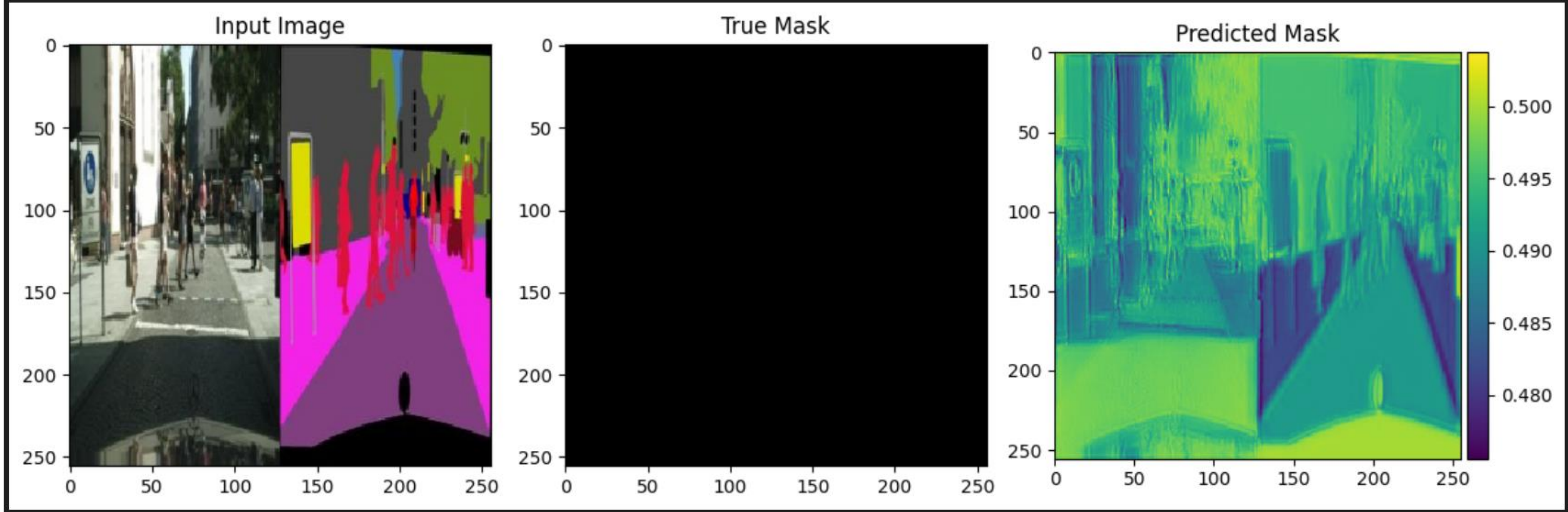
Algorithm Selection:

- **Data Characteristics**:
  - High-resolution images with detailed annotations.
  - Diverse urban driving scenes with varied objects and environmental conditions.
- **Why U-Net**:
  - **Accuracy:** High effectiveness in image segmentation, offering detailed segmentation maps.
  - **Efficiency:** Suitable for real-time applications due to computational efficiency.
  - **Versatility:** Handles variability in real-world scenarios, capturing fine-grained details and context.
- **Proven Success**: Robust and reliable in complex segmentation tasks across different fields.

- **Data Input:**
  - This dataset has 2975 training images files and 500 validation image files. Each image file is 256x512 pixels, and each file is a composite with the original photo on the left half of the image, alongside the labeled image (output of semantic segmentation) on the right half.
- **Training Process**:
  - Used the data augmentation to create more data and used dropout layer additionally in the architecture.
  - Compiled using the adam optimizer and loss function taken as binary cross-entropy.
- **Prediction Process:**
  - The trained algorithm takes the input images and it is passed through the virtual true mask by U-net model to predict the actual mask segmented for the self-driving cars.

Result

```
1/1 [==============================] - 0s 349ms/step
```




```
16/16 [==============================] - 166s 10s/step - loss: 0.6805 - accuracy: 0.9921
Validation Loss: 0.6804660558700562
Validation Accuracy: 0.9920607209205627
```

Project Link

- https://github.com/rajgoyal1729/Image-Segmentation-for-self-driving-cars

Conclusion

- **Summary of Findings and Effectiveness of the Proposed Solution**
- **Findings:**
- **Accuracy:** The U-Net model provided high accuracy in segmenting different elements (pedestrians, vehicles, road signs, lane markings) from the Cityscapes dataset.
- **Efficiency:** The model demonstrated efficiency in processing high-resolution images, suitable for real-time applications.
- **Robustness:** U-Net was able to handle the complexity and variability of urban driving scenes, maintaining consistent performance across different conditions.
- **Effectiveness:**
- **Precise Segmentation:** The detailed segmentation maps produced by U-Net contributed to the precise identification and classification of various elements in driving scenes.
- **Real-Time Application:** The computational efficiency of U-Net ensured that it met the requirements for real-time image segmentation, which is crucial for self-driving cars.
- **Improved Safety:** By accurately identifying and classifying elements in the driving environment, the model enhances the safety and reliability of autonomous driving systems.

- **Challenges Encountered**

1.Memory Constraints:

1. Handling high-resolution images from the Cityscapes dataset led to memory errors.
2. Solution: Implemented data augmentation and mini-batch training to manage memory usage.
3. The dataset does not consist of a separate label folder for all the corresponding images.
4. Solution: Had to generate synthetic labels for all the images respectively.

2.Training Time:

1. The training process was time-consuming due to the complexity of the U-Net architecture.
2. Solution: Used hardware acceleration (e.g., GPUs) to speed up the training process.

Future Scope

- **Potential Improvements**

1. **Enhanced Data Augmentation:**
    1. Implement more sophisticated augmentation techniques to increase the diversity of training data and improve model robustness.
2. **Model Optimization:**
    1. Use model pruning and quantization to reduce the model size and improve inference speed without compromising accuracy.
3. **Advanced Architectures:**
    1. Explore more advanced segmentation architectures like DeepLab or SegNet for potentially better performance.
4. **Transfer Learning:**
    1. Utilize pre-trained models on similar tasks to reduce training time and improve initial performance.

Importance

- **Importance of Image Segmentation for Self-Driving Cars**
- **Obstacle Detection:** Precise segmentation is crucial for identifying obstacles (e.g., pedestrians, vehicles) to ensure safe navigation.
- **Traffic Sign Recognition:** Accurate segmentation of traffic signs and signals is essential for obeying traffic rules and ensuring safety.
- **Lane Marking Detection:** Reliable segmentation of lane markings helps in maintaining the correct lane and assists in lane-keeping systems.
- **Environmental Understanding:** Comprehensive segmentation of the driving environment (e.g., roads, sidewalks, buildings) improves the overall situational awareness of autonomous vehicles.

References

- Dataset: https://www.kaggle.com/datasets/dansbecker/cityscapes-image-pairs/data