Progetto di Ingegneria Informatica

Data Augmentation Solution for Advanced Computer Vision

POLITECNICO
MILANO 1863

Menelik Nouvellon
10916530

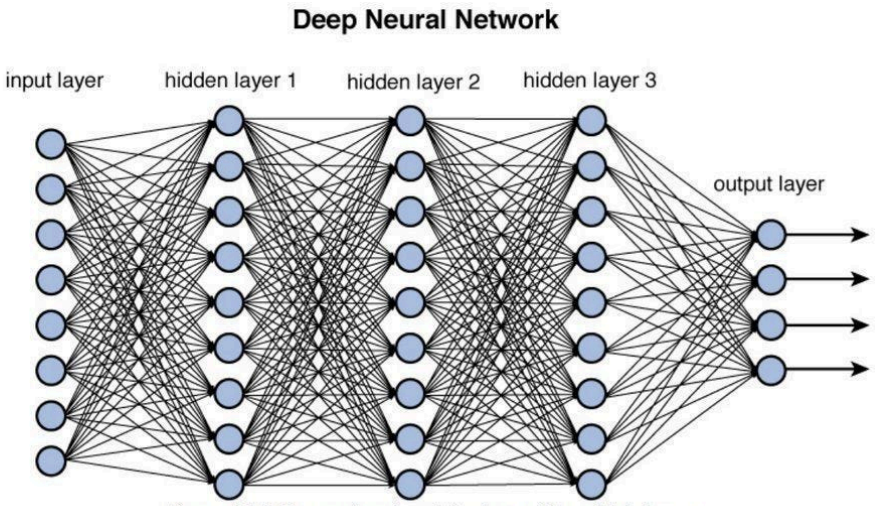
Anno Accademico 2022/2023

Tutor : Davide Yi Xian Hu

Introduction

Deep Learning and Its Applications in Computer Vision

Deep Learning, a subset of artificial intelligence, focuses on training and utilizing artificial neural networks with multiple layers, simulating the human brain's functionality. It has shown significant effectiveness in a wide range of applications, particularly in the field of computer vision.



Computer vision encompasses the understanding and interpretation of images and scenes by computers and software systems. Deep learning has revolutionized this field by enabling computers to perform tasks that were previously challenging, such as object detection, semantic segmentation, and depth estimation.

One prominent application of deep learning in computer vision is tumor detection in medical imaging. By recognizing patterns, shapes, and textures in imaging data, deep learning models can detect and classify tumors, aiding in early diagnosis and improved treatment outcomes.

Another key application is in the realm of autonomous driving. Deep learning algorithms enable computer systems to analyze vast amounts of data from various sensors and make informed decisions, thereby navigating safely and efficiently in the real world. This involves identifying other vehicles, pedestrians, and road signs, and deciding on actions such as steering, braking, and maneuvering.

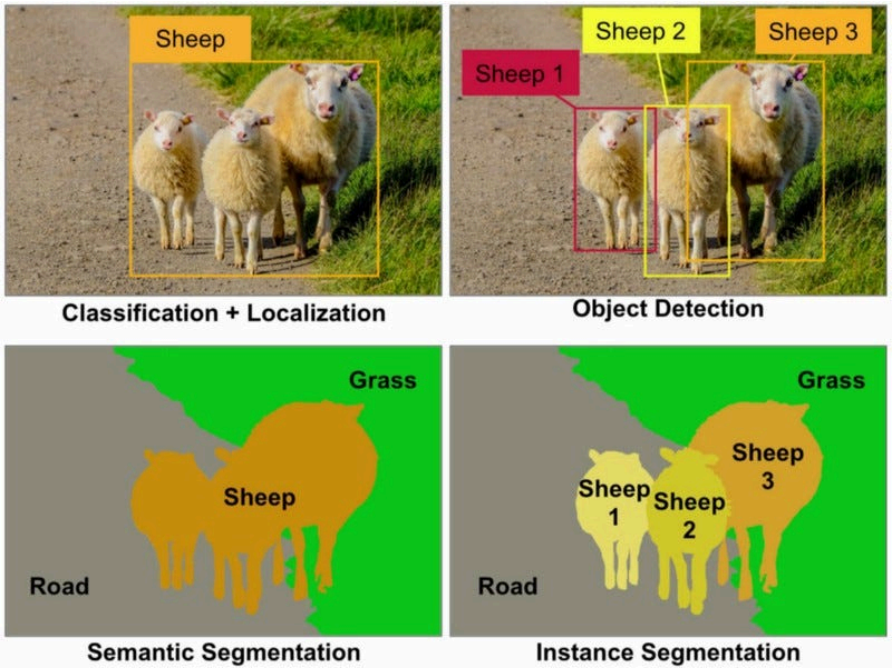
In essence, the advent of deep learning has significantly boosted the capabilities of computer vision systems, opening up new possibilities and applications.

Semantic Segmentation

Semantic segmentation is one of the key problems in the field of computer vision. Looking at an image, there's more to understand than just what individual objects are present - the location, shape, and spatial relationships among these objects are also significant pieces of information. This is where semantic segmentation comes in, as it not only identifies the objects present in an image, but also discerns where exactly in the image they are, by assigning each pixel of the image to a particular class.

In simple terms, semantic segmentation is the practice of labeling each pixel in an image with a class of objects. It differs from image classification, where the task is to predict a single class label

for the entire image, or object detection, where the goal is to predict bounding boxes around certain objects in the image.



Semantic segmentation finds application in numerous fields such as autonomous driving, robotics, industrial automation, medical imaging, and augmented reality among others. It plays a crucial role in enabling machines to perceive their surroundings in a manner that is closer to human visual perception.

However, training robust semantic segmentation models often requires a large amount of annotated training data. Acquiring such data can be time-consuming and expensive, especially considering that the annotation process often requires manual labor from subject matter experts.

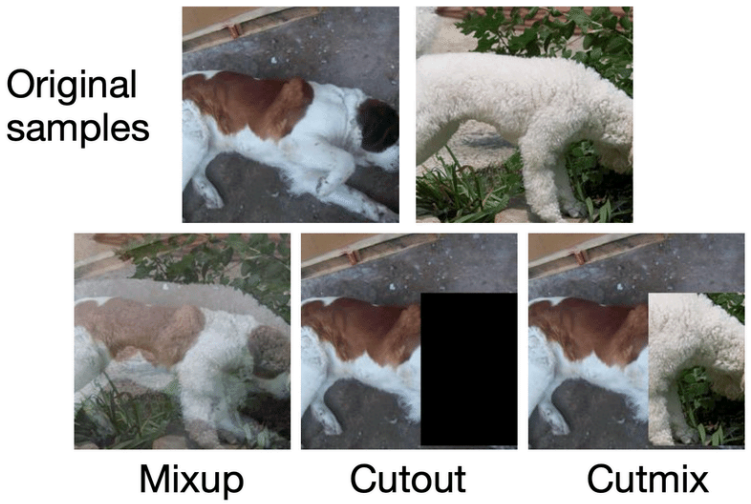
This project explores the use of data augmentation as a solution to this challenge.

Data Augmentation

Data augmentation is a powerful strategy used in machine learning, which enables us to increase the diversity of data available for training models, **without actually collecting new data**. It involves creating new training samples by applying various kinds of perturbations on the original dataset, which are reflective of the real-world variations the model might encounter. By augmenting the data in this way, we can essentially generate a more robust and diverse dataset, which can lead to the training of a model that is **capable of generalizing better to unseen data**.

Data augmentation is particularly beneficial in the field of computer vision, where image data can be manipulated via a number of ways without losing their labels. Some common image data augmentation techniques include **rotation, scaling, translation, flipping, and cropping**. By applying these transformations, we essentially provide the model with a wider variety of perspectives, which helps it to understand the data better and reduces the chances of overfitting.

In addition to these traditional techniques, there are also more complex and advanced data augmentation methods that have been recently introduced, such as CutMix, Mixup, and Cutout.



CutMix is an augmentation strategy that patches cut portions from one image onto another. The ground truth labels are also mixed proportionally to the area of the cuts. This results in a model that can localize better and is less likely to focus on the background or other irrelevant parts of an image.

Mixup performs data augmentation in the input space by taking a convex combination of two inputs and their corresponding labels. This has been found to regularize the model and help it generalize better to unseen data.

Cutout is another technique that regularizes the model by randomly masking out one or more patches from an image during training. This encourages the model to take into account the entire context of the image rather than relying on a small number of discriminative regions, thus leading to improved robustness and generalization.

In the scope of this project, these data augmentation techniques are used and compared to see how they impact the performance of semantic segmentation models.

Project

Purpose of the Project

The primary objective of this bachelor project is to examine the effects of various data augmentation techniques on the performance of semantic segmentation models. This is achieved by implementing and comparing the impact of different data augmentation strategies.

Our approach involves applying these techniques to the SHIFTDataset (https://www.vis.xyz/shift/), an autonomous driving task dataset, and observing the resulting changes in model accuracy, generalizability, and robustness. This dataset, which is represented as a torch matrix of pixels in the RGB color space, provides a comprehensive platform for this exploration due to its diversity and complexity, including various real-world environmental conditions, sensor modalities, perception tasks, and continuous domain shifts, challenging models to adapt and perform robustly in complex driving scenarios.

By comparing these different methods, we aim to not only understand the individual merits and potential drawbacks of each technique, but also identify whether a combination of these techniques can lead to even better model performance. In doing so, we hope to contribute to the current

understanding of how data augmentation can be effectively used to improve semantic segmentation.

Furthermore, the results of this project could guide the development of future computer vision applications. By understanding the impacts of various data augmentation techniques, developers could more effectively train their models, thus reducing the time, costs, and data requirements associated with manual annotation.

Ultimately, our goal is to push the boundaries of what is currently achievable in semantic segmentation, paving the way for more advanced, accurate, and efficient computer vision systems in the future.

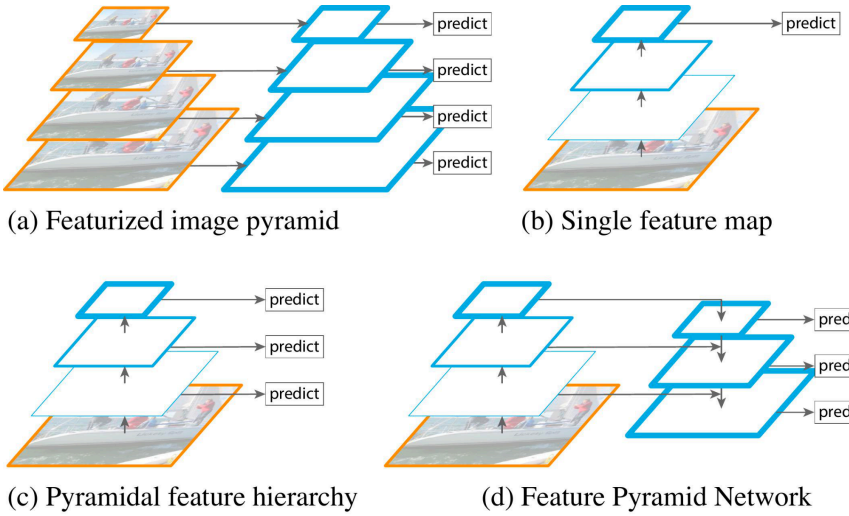
Libraries and Dataset

PyTorch

In this project, we leverage the powerful functionalities of PyTorch (https://pytorch.org/), an open-source machine learning library widely used for applications such as computer vision and natural language processing. PyTorch's dynamic computational graph and a rich set of tools and libraries are particularly useful for tasks like semantic segmentation. We especially capitalize on PyTorch's data loading and processing capabilities for efficient and customizable handling of the SHIFTDataset.

In addition to leveraging the powerful functionalities of PyTorch, we also utilize PyTorch Lightning (https://www.pytorchlightning.ai/), a lightweight PyTorch wrapper that further simplifies the training and development process. PyTorch Lightning provides a high-level interface and pre-defined patterns for common tasks, allowing us to abstract away low-level details and focus on the core aspects of our models. It offers advanced features such as automatic optimization, distributed training, and model checkpointing, enhancing productivity and scalability in our project. By integrating PyTorch Lightning into our workflow, we can take advantage of its streamlined and efficient approach to build and train models on the SHIFTDataset.

The Feature Pyramid Network (FPN) is a neural network architecture used for semantic segmentation. It creates a pyramid of feature maps with different resolutions by combining a top-down pathway and lateral connections. The top-down pathway upsamples higher-level features and merges them with lower-level features through lateral connections, allowing the network to capture both fine-grained details and high-level semantic information. This multi-scale representation enhances the network's ability to accurately segment objects of different sizes in the SHIFTDataset.



(a) Featurized image pyramid (b) Single feature map

(c) Pyramidal feature hierarchy (d) Feature Pyramid Network

SHIFT Dataset

The SHIFTDataset is a synthetic driving dataset designed explicitly for continuous multi-task domain adaptation in autonomous driving systems. It provides a comprehensive sensor suite and annotations for various mainstream perception tasks related to autonomous driving. This allows us to closely simulate the mutable nature of real-world environments, capturing both discrete and continuous shifts in environmental factors, including cloudiness, rain and fog intensity, time of day, and vehicle and pedestrian density.

The SHIFTDataset includes a range of sensor data, such as multi-view RGB cameras, stereo RGB cameras, depth cameras, optical flow cameras, GNSS/IMU sensors, LiDAR point clouds, and their annotations. The dataset is represented as a torch matrix of pixels in the RGB color space, providing high-quality input for our models.

The SHIFTDataset classifies every pixel of an image in a range of 23 classes. The 23 possible classes are the following: unlabeled, building, fence, other, pedestrian, pole, road line, road, sidewalk, vegetation, vehicle, wall, traffic sign, sky, ground, bridge, rail track, guard rail, traffic light, static, dynamic, water and terrain.

Our project focuses particularly on the aspect of semantic segmentation, which is one of the twelve mainstream perception tasks covered by the SHIFTDataset.

SHIFT DatasetWrapper

To integrate data augmentation techniques directly into the training process, we developed a `SHIFTDatasetWrapper`. This wrapper class extends the original SHIFTDataset, providing an interface to apply our chosen data augmentation strategies directly and automatically during the data loading pipeline.

Each epoch during the training process sees the application of different transformations on the images before they are fed into the model. This approach serves a dual purpose: it not only exposes the model to a broader variety of data during training but also helps reduce overfitting by presenting the model with 'new' data at each iteration.

The combination of PyTorch and the `SHIFTDatasetWrapper` offers an efficient and flexible pipeline for conducting experiments with various data augmentation techniques and evaluating their impact on semantic segmentation.

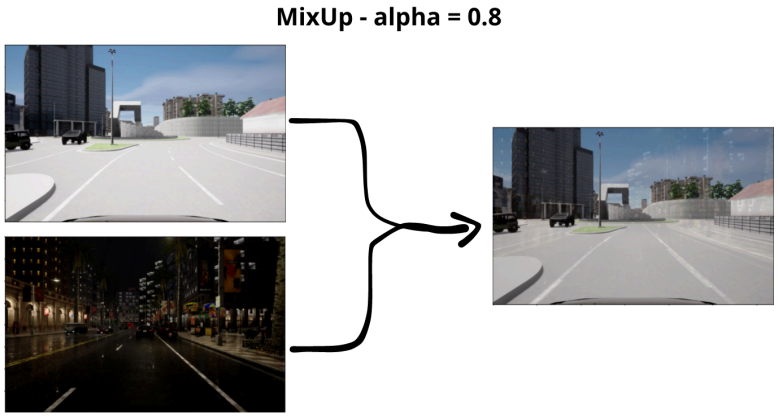
Techniques Implemented

The data augmentation techniques implemented in this project can be broadly categorized into two groups: those for mixing images and those for mixing ground truth images.

Mixing Images

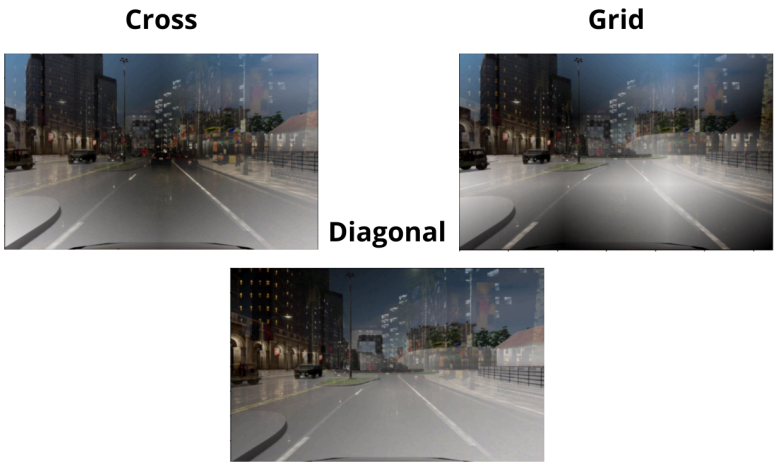
Three key techniques were developed and implemented for the purpose of mixing images:

1. **Mixup**: Mixup performs data augmentation by taking a convex combination of two images and their corresponding labels. It operates in the input space and generates virtual training examples. By presenting these diverse examples to the model during training, it aids in enhancing the model's generalizability.



2. **Fade**: The Fade technique is a more sophisticated form of data augmentation that involves blending two images either horizontally or vertically. This results in an image that smoothly transitions from one image to another across the specified axis. It helps the model learn to handle different transitions in the input data, contributing to improved model robustness.

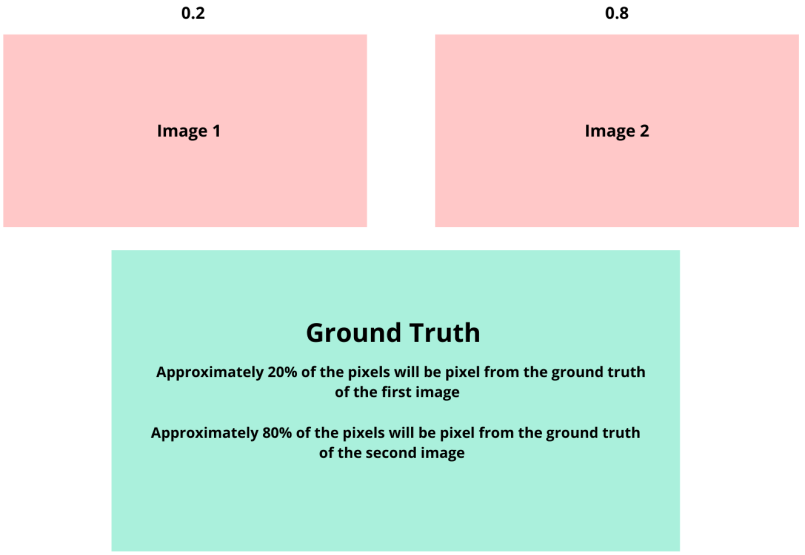
3. **Others**: In addition to Mixup and Fade, we also explored more complex blending techniques, namely Diagonal, Cross, and Grid. The Diagonal technique mixes one horizontal and one vertical fade image. Cross mixes one horizontal and one vertical fade image, with two fades applied, while Grid uses a similar approach but applies three fades. These techniques provide the model with even more diverse training examples, helping to ensure that the model can generalize well to unseen data.



Mixing Ground Truth Images

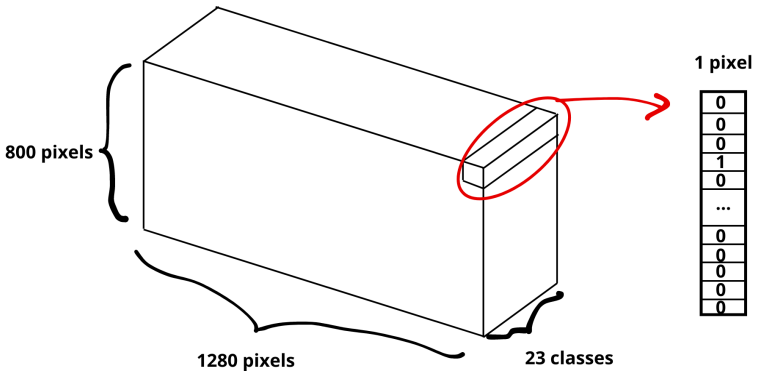
1. **Mixing Using Probability**: In this method, the class for each pixel is chosen based on a random value determined by the class probabilities of the same pixel in two images. This results in a more

probabilistic and varied representation of the ground truth, promoting the learning of a model that can adapt to varied scenarios.



2. **Mixing Using One Hot Encoding**: The second method involves the use of One Hot Encoding, a process of converting categorical data into a form that could be provided to ML algorithms to improve predictions.

We used three-dimensional tensor matrices of size (23, 800, 1280) to represent the 23 possible classes.



In this way, for every pixel, we can keep the information of the mixing by keeping both classes in the 23-size tensor matrix.

Experiment and results

Experimental setup

In our experimental setup, we utilized several settings to conduct the experiments on the SHIFTDataset. Here are the details:

1. Early Stopping: We employed early stopping to prevent overfitting and determine the optimal training stopping point. The early stopping callback monitored the validation loss, and if there was no improvement for two consecutive epochs, training was stopped.

2. GPU Acceleration: To accelerate the computation and handle the resource-intensive nature of the experiments, we utilized GPUs (Graphics Processing Units). GPUs are well-suited for deep learning

tasks as they can parallelize computations and significantly speed up training and inference processes.

3. Training and Test Split: We split the SHIFTDataset into a training set and a test set. The training set constituted 90% of the data, while the remaining 10% formed the test set. This split ensured that the model was trained on a majority of the data and evaluated on a separate unseen portion to assess its generalization capabilities.

4. Batch Size: During training, we used a batch size of 8. Batch size refers to the number of samples processed by the model in each forward and backward pass. A larger batch size can improve training efficiency but requires more memory. We chose a batch size of 8 as a balance between computational efficiency and memory constraints.

5. Maximum Epochs: We set the maximum number of epochs to 20. An epoch represents a complete pass through the entire training dataset. Training for a fixed number of epochs ensures that the model has an opportunity to learn from the data while preventing overfitting or excessively long training times.

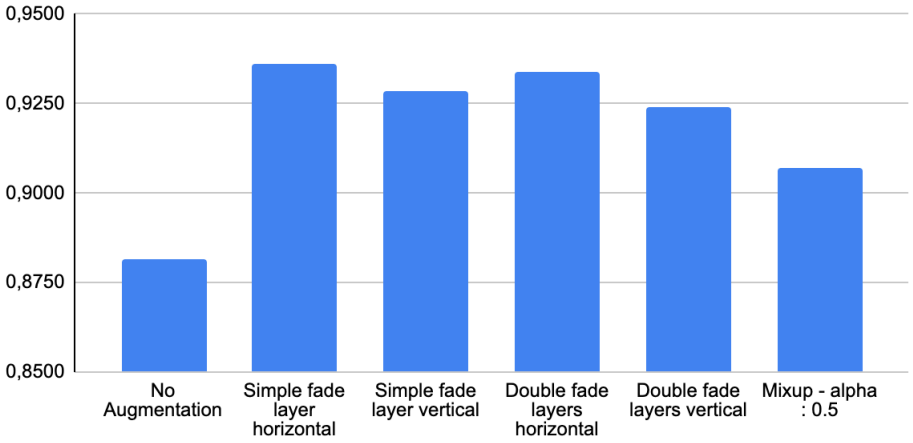
Overall, each model was trained for 20 epochs, with early stopping triggered if there was no improvement in validation loss for two consecutive epochs. The training process, performed on GPUs, required a total of 144 hours to complete.

These settings allowed us to train and evaluate the models effectively on the SHIFTDataset, considering computational resources, generalization capabilities, and the prevention of overfitting.

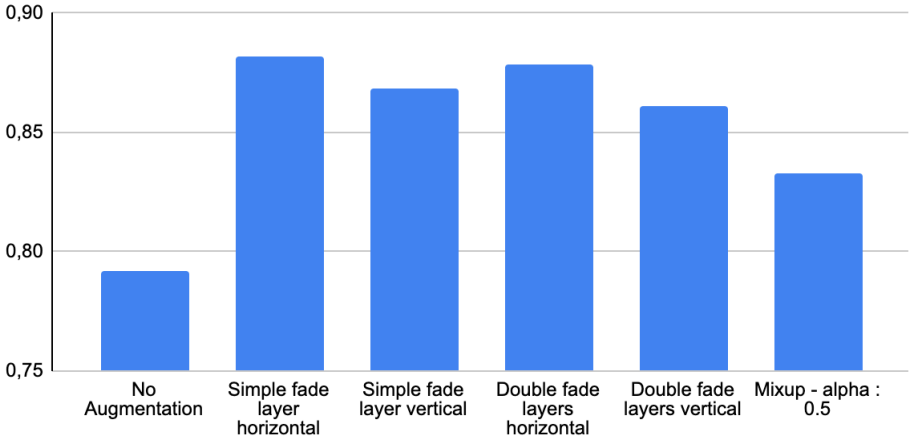
Results

After running all the models (using as ground truth mixing technique **only the probability one**), we got some interesting scores. We obtained multiple metrics and loss functions but we'll be presenting 3 main scores : F1 score, F2 score and Dice loss score.

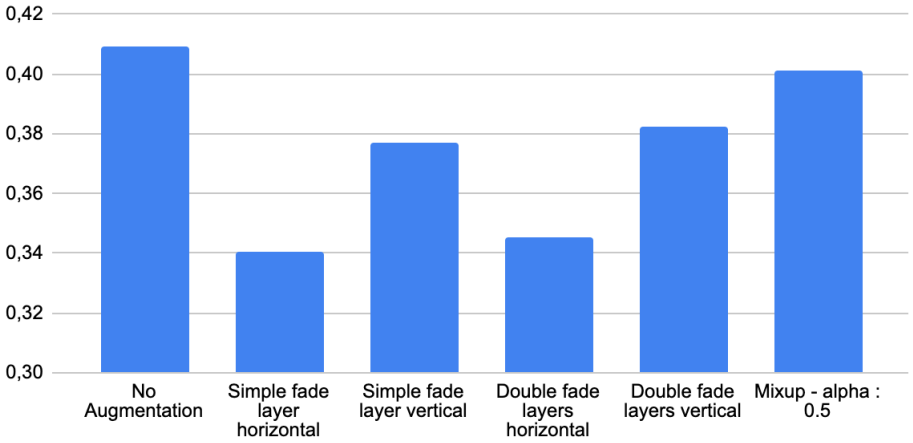
Metric/Loss	Definition	Purpose	Evaluation
F1 Score	Harmonic mean of precision and recall	Evaluates overall model performance in classification tasks	0 - 1 : the highest value the best
F2 Score	Extension of F1 score, emphasizes recall	Prioritizes correctly identifying positive instances	0 - 1 : the highest value the best
Dice Loss	Measures dissimilarity between predicted and ground truth masks	Evaluates accuracy of predicted segmentation masks in image segmentation tasks	0 - 1 : the lowest value the best



F1 score



F2 score



test_dice_loss score

Observations and Conclusions

1. **Efficacy of Data Augmentation:** It's clear that data augmentation had a substantial impact on the performance of the semantic segmentation models. Compared to the baseline model (without data augmentation), all of the implemented augmentation techniques improved the performance metrics. Additionally, all of the data augmentation techniques contributed to reductions in the various loss metrics.

2. **Superiority of Fade Techniques:** Notably, the Fade technique seemed to yield superior results compared to the Mixup technique. This could be due to the fact that the Fade technique introduces more spatial variance into the training data, encouraging the model to learn more generalized features.

3. **Direction of Fade:** The number of Fade layers and their direction (horizontal or vertical) also played a crucial role in determining the model's performance. Both simple fade horizontal and double fade horizontal techniques yielded higher scores on the performance metrics compared to their vertical counterparts. This indicates that our model may be more sensitive to horizontal variations in the images.

4. **Mixup Technique:** The Mixup technique, though not as effective as the Fade techniques in this project, still showed significant improvement over the baseline model without data augmentation. It showcases the potential of mixup as a simple yet effective technique for increasing the diversity of training data.

5. **Performance Metrics:** Overall, all of the implemented data augmentation techniques improved the performance of the semantic segmentation models. However, the degree of improvement varied, which demonstrates the importance of experimenting with different data augmentation techniques to identify which one best suits the given task.

Conclusion

The task of semantic segmentation plays a critical role in a variety of computer vision applications, from autonomous driving to medical image analysis. It presents unique challenges that require advanced machine learning techniques to overcome. This project explored one such technique – data augmentation – and its impact on semantic segmentation models.

In this work, we conducted extensive experiments to assess the impact of various data augmentation techniques, including Mixup, Fade, and several others. We utilized the comprehensive SHIFTDataset, exploiting its synthetic driving environment data to test our methods. Our experiments spanned various techniques and dimensions, from mixing images to mixing ground truth images, which were three-dimensional tensor matrices representing multiple classes.

Our findings indicate a clear enhancement in model performance when employing data augmentation. Particularly, Fade techniques, especially when applied horizontally, demonstrated significant improvements in the performance metrics. The direction and number of layers in the Fade technique emerged as crucial factors influencing the results.

While our experiments have provided valuable insights into the utility of data augmentation in semantic segmentation, they also open avenues for further exploration. The combination of different techniques, finer adjustments to parameters in existing methods, or the development of entirely new augmentation strategies are all promising areas for future work.

In conclusion, data augmentation has proved to be an effective strategy for improving the performance of semantic segmentation models. This work contributes to the growing body of research emphasizing the importance of data augmentation in developing robust and reliable computer vision systems.