Machine Learning Project 02

Xun Zhang
Haotian Shen
Westlake University

***Abstract*—In this project, we implemented linear regression, CNN and FCN variants, as well as the attention-based Segformer to perform segmentation on RGB images containing roads and buildings. We analyze the strengths and weaknesses of different architectures in semantic image segmentation, focusing on the performance of CNNs with different strides and FCNs with various skip connection structures. Finally, we selected the architecture with the highest accuracy in the test set and used it to generate predictions for 50 test images.**

I. Introduction

Semantic image segmentation is a fundamental task in computer vision, aiming to assign a semantic label (such as road, building, or sky) to every pixel in an image. With the advancement of deep learning, this field has made remarkable progress over the past decade. Convolutional Neural Networks (CNNs), which first demonstrated strong performance in image classification, have since been widely applied to segmentation tasks.

In 2015, Long et al. proposed Fully Convolutional Networks (FCNs), the first architecture to adapt traditional classification networks for dense pixel-wise prediction. This work marked the beginning of deep learning-based semantic segmentation. FCNs replaced fully connected layers with convolutional layers to preserve spatial information and introduced skip connections to fuse features across multiple levels, significantly improving segmentation accuracy. Subsequently, a variety of CNN-based architectures—such as U-Net and the DeepLab family—were developed, further advancing the field.

Although CNNs are excellent at capturing local features, they are inherently limited in modeling global context. To address this, Transformer-based architectures have been introduced to segmentation tasks in recent years. Compared to early vision Transformers such as ViT and SETR, which offer powerful global modeling capabilities, these models often suffer from large parameter sizes, high computational costs, and reliance on positional encodings.

In 2021, Segformer, introduced by NVIDIA and the University of Hong Kong, tackled these issues by adopting a lightweight hybrid Transformer encoder and a multi-scale feature aggregation module. Unlike SETR or Swin Transformer, Segformer does not rely on positional encodings or heavy upsampling modules, yet achieves higher accuracy and faster inference. It has become one of the most efficient and accurate segmentation models to date.

In our project, we experimented with several of these classic architectures. Using logistic regression as a baseline, we implemented CNN, FCN, and Segformer models to perform binary segmentation on each 16×16 patch of 608×608 RGB images. We found that deeper skip connections significantly improved the performance of FCNs,achieving up to 86% accuracy. Moreover, using a stride size equal to the patch size 16*16 in the first convolution layer led to decrement of the performance.For FCN, we analyzed the role of skip connections and the impact of different initial convolution layer strides. Finally, applying the Segformer model enabled us to exceed 92% accuracy and 71.43% MIoU, We compared the images generated by FCN8s and SegFormer, highlighting the different focuses of convolution and attention mechanisms, and illustrating the advantage of attention mechanisms in capturing long-range information in image semantic segmentation.

II. Method

A. Data Processing

We noticed that the images in the test set (608×608) and the training set (400×400) differ in size. Therefore, we resized the training images from 400×400 to 608×608 using bilinear interpolation for the input images and nearest-neighbor interpolation for the ground truth labels. The resized results were saved in the training1 folder.

B. Models

1) Logstic Regression Model: We divide the ground truth images into 16×16 patches. For each 16×16 patch, we use the (mean, var) of that patch as the input X_i, and the category of that patch (1 represents ground, 0 represents house) as Y_i, then perform logistic regression on X_i and Y_i.
2) CNN_simple: We follow the same approach as in regression, applying three convolution layers and one fully connected layer to each 16×16 patch. Since the ratio of roads to houses in the images is approximately 7:3, we use weighted cross-entropy as the loss function to improve performance.
3) FCN8s_simplified: We constructed the structure shown in Figure 1. Here, we set stride=1 and kernel size=1 for the first convolution, which means our convolution layer captures more detailed global information.

4) FCN8s: we built the structure shown in Figure 2.Here, we add two skip connection
5) FCN8S_stride16_simplified: We use the structure shown in Figure 3.In the first convlution,We use stride=16 to capture the information between patches.
6) FCN8S_stride16: We use the sturcture shown in Figure 4.We add two skip connection compared to above model.
7) Segformer: we choose pretrained Segformer from hugging face 'nvidia/segformer-b2-finetuned-ade-512-512' with the pipeline shown in Figure 5 and perform fine-tuning on our trainning set.

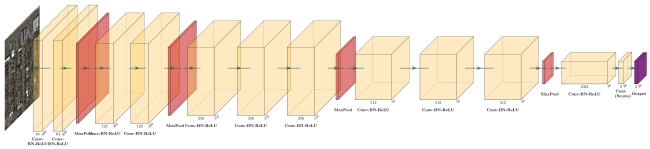

Figure 1. FCN8s_simplified*

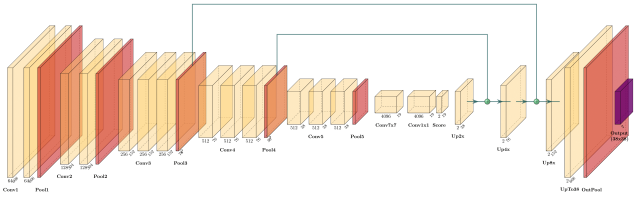

Figure 2. FCN8s*

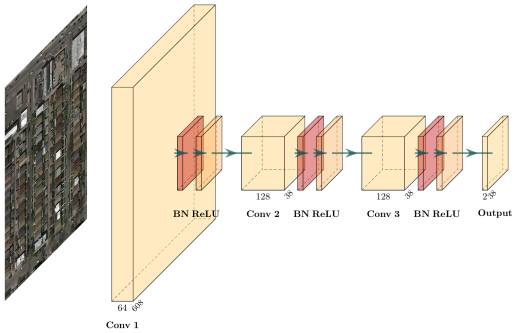

Figure 3. FCN8S_stride16_simplified*

III. METHODS TEST

A. Traning Setup&Hyperparameter

We randomly select 80% of the data from the "training1" folder as the training set and use the remaining 20% as the test set.

Data_augmentation: We randomly choose 10% images from training_set, and flip it or rotate it with both 50% probability.

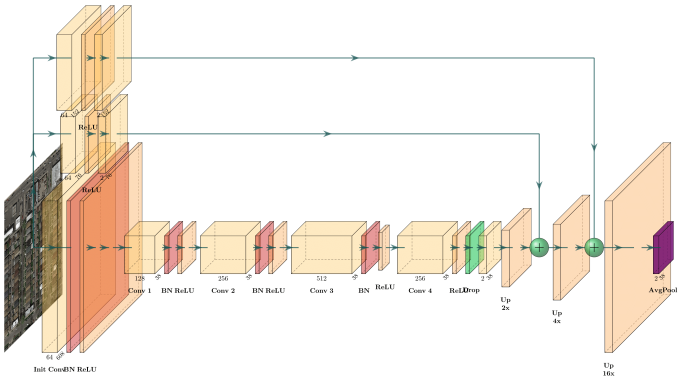

Figure 4. FCN8S_stride16*

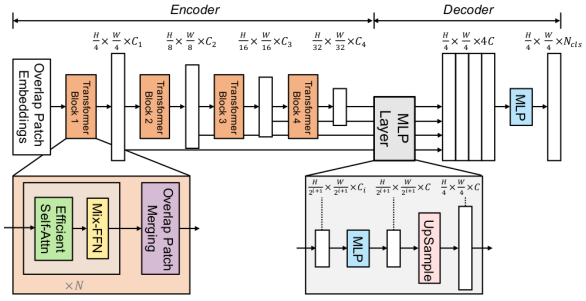

Figure 5. Segformer*

Optimizer: Adam
Learning rate: 0.001
Loss: Cross Entropy Loss
batch_size=4
num_epoches=

- 4: CNN_simple,FCN8s_simplified,
- 6: FCN8s,FCN8S_stride16_simplified, FCN8S_stride16
- 10: Segformer

The final result is saved in **submission.csv**, and you can reproduce it by running **main.py** or **Segformer.py**

B. Performance

Model	Acc(%)	Data_aug Acc(%)	MIoU(%)
logistic regression	54.73	—-	—-
CNN_simple	75.38	69.66	50.89
FCN8s_simplified	79.47	81.17	40.49
FCN8s	86.30	82.19	50.61
FCN8S_stride16_simplified	80.93	81.18	32.11
FCN8S_stride16	78.62	79.89	29.96
Segformer	92.32	—-	71.43

Table I
PERFORMANCES FOR DIFFERENT MODELS

After training, we evaluate the accuracy of the test set and that with data augmentation. Besides, we use MIoU

*Note that Figure 1 $\sim$ 4 are generated by PlotNeuralNet and2Figure 5 is from SegFormer paper

by calculating the average Intersection over Union (IoU) across all semantic classes.MIoU averages the performance across different categories, avoiding the situation where some categories perform very well while others perform poorly. This is particularly important for datasets with class imbalance.

C. Analysis

- Using only the mean and variance as features for each patch in **logistic regression** results in limited representational capacity, leading to relatively low accuracy.
- We observed the same issue with **CNN_simple**: If only local-level features are extracted from patches, the model struggles to achieve satisfactory classification performance. This is because, in cases where certain rooftops have colors similar to roads, our CNN fails to effectively distinguish between them.
- In our experiments, we initially attempted pixel-level semantic segmentation on 608×608 images using FCN8s directly. However, the output resembled random noise, and even after manually setting a threshold, the results were unsatisfactory. Therefore, we added a **pooling** layer for FCN8s and FCN8S_stride16, generating a 38×38 segmentation map instead.
- FCN8s_simplified uses convolutional blocks as the encoder, while FCN8s adds **a decoder part** that uses deconvolution (transpose convolution) to recover the micro information lost during the convolution process. This results in a significant performance improvement for FCN8s
- **Skip connections** combine information from deep convolution layers and shallow convolution layers, enabling the model to capture both global and local information effectively. Similar to ResNet, these skip connections also facilitate the backpropagation of gradients, enhancing training stability and efficiency.
- In FCN8S_stride16_simplified and FCN8S_stride16, we aimed to better capture the overall information within each patch. Therefore, we tried setting the **initial stride to 16**, which matches the size of the patches. This indeed improved the model's runtime speed, but unfortunately, we did **not achieve higher accuracy**.
  In FCN8S_stride16, because the initial stride is large, we tried to directly extract the initial image for the skip connection, but the results were still **not particularly good**. Even though we obtained a larger receptive field, the local micro-geometric information is crucial for classifying roads and houses. For example, if a car is detected within a patch, and many trees around the patch, it is almost certain that the patch belongs to the road.
- **SegFormer**, SegFormer, based on the transformer architecture, can better capture global information. For example, each patch is part of a road or a house, and roads or houses often satisfy certain geometric structures. Compared to the convolution layers of CNNs, SegFormer's attention mechanism can better preserve this overall information, **leading to the highest accuracy**. In the comparison of Figure 6 and Figure 7, we can clearly see that SegFormer can predict an entire road very well even the road is very narrow, with a very smooth prediction map, while FCN8s has many scattered small patches.
- In our tests, **data augmentation did not improve** the model's performance. This is because both the training set and the test set consist of images taken from relatively fixed heights and angles. Therefore, transformations such as flipping and rotation do not significantly contribute to enhancing the model's performance by supplementing the training set.

D. Training Method

We have to choose methods with both high accuracy and MIoU to have good performance in test set. So we choose the last combination which uses all the methods.

IV. Possible Improvement

Figure 6. Road recognition from FCN.

Figure 7. Road recognition from Segformer

- The provided training set contains only 100 images, which may not be sufficient for the model to learn effectively. We can look for additional datasets to perform segmentation training.
- We found that there is a lot of noise in the images, such as cars on the road, trees on the roadside, and some courtyards that resemble roads. These areas are highly likely to cause misclassifications by the model. For example, when the architecture learns that roads typically have cars, it may misclassify some parking lots with very similar color as roads, as shown in Figure 6. FCN8s misclassified parking lots as roads. Therefore, we believe that adding more categories could help improve the accuracy of segmentation.
- Simple augmentations such as flipping, rotation, and scaling are not well-suited for this task. In future work, we can try adding Gaussian noise, brightness transformations, GridMask, and multi-sample data augmentation methods to improve the model's stability and generalization ability.
- Our dataset suffers from class imbalance (e.g., building pixels are far more prevalent than road pixels). To avoid the model's bias toward the majority class, we can use weighted loss functions such as weighted cross-entropy or focal loss. However, we only implemented this approach in the CNN_simple model.
- Due to the limitations of computational resources, we did not perform hyperparameter tuning.

V. Ethical Risks

A primary ethical risk to consider is the project's impact on public Welfare, stemming from the potential consequences of model errors. Our analysis shows that models can make significant mistakes, such as an FCN misclassifying a parking lot as a road because its color was similar to that of a real road. In a real-world application, such as autonomous navigation or city planning, this type of error could lead to dangerous accidents or incorrect allocation of infrastructure funds. Furthermore, the technology's core function of identifying infrastructure could be repurposed for harmful uses, such as mass surveillance or military targeting, posing a direct risk to public safety.

Another significant ethical concern is Privacy, which arises from the nature of the data being processed. The model is trained on aerial RGB images to identify roads and buildings. While the project used a provided training set of 100 images, the use of high-resolution aerial imagery in a real-world deployment could capture sensitive details of private property, vehicles, or even individuals without their consent. The project did not specify methods for data protection, such as anonymization or encryption. If this system were scaled, the collection and storage of such data would pose a substantial privacy risk to the people living in the surveyed areas.

VI. summary

In this project, we compared the performance of different architectures in image semantic segmentation, analyzed their advantages and disadvantages, and ultimately selected SegFormer as the final model to make predictions on the test set.

VII. Appendix

1) Code is available at our Github repository.
2) Digital Ethics Canvas is attached in the next page:

DIGITAL ETHICS CANVAS

CONTEXT

SOLUTION

BENEFITS

Providing an automated and highly accurate method for segmenting roads and buildings from aerial images.

WELFARE

RISK

- Can the solution be used in harmful ways, in particular with regards to vulnerable populations?
- What kind of impacts can errors from the solution have?
- What type of protection does the solution have against attacks or misuse?
- Answer:
- 1. The model, which identifies buildings and roads, could be used for surveillance by governments or private entities. In a military context, it could be used for targeting infrastructure.
- 2. Errors, such as misclassifying parking lots as roads, could lead to flawed urban planning or accidents if used for autonomous navigation.
- 3. As a student research project, the solution does not have specific protections against misuse or attacks built into it.

MITIGATION

For a real-world deployment, access controls and user agreements that explicitly forbid use for surveillance or military applications would be necessary.

FAIRNESS

RISK

- How accessible is the solution?
- What kinds of biases may affect the results ?
- Can the outcomes of the solution be different for different users or groups?
- Could the solution contribute to discrimination against people or groups?
- Answer:
- 1. The solution requires technical expertise to run and is not easily accessible to the general public.
- 2. The model is trained on a small set of 100 images from "relatively fixed heights and angles," creating a high risk of bias against different architectural or geographic styles. There is also a class imbalance in the dataset.
-

MITIGATION

The user may capture images with different sizes, which may conflict with the pretrained models, causing errors.

AUTONOMY

RISK

- Can users understand how the solution works and what its limits are?
- Are users able to make choices (e.g. consent, settings) in their use of the solution and how?
- How does the solution affect user autonomy and agency?
- Answer:
- 1. Non-technical users would likely not understand the model's limitations or the reasons behind specific errors, such as misclassifying a rooftop similar in color to a road.
- 2. The developers make choices regarding the model, but an end-user of the final segmentation map has no ability to adjust settings or provide consent.
- 3. If a decision-maker relies on the tool's output without understanding its flaws, their ability to make an informed decision is diminished.

MITIGATION

Any deployed version of this tool can include a confidence score for its predictions and highlight areas where classification certainty is low.

PRIVACY

RISK

- What data does the solution collect
- Is it collecting personal or sensitive data
- Who has access to the data?
- How is the data protected?
- Could the solution disclose / be used to disclose private information?
- 1,2: The solution processes RGB aerial images; if the resolution is high enough, these could contain sensitive information about private property or individuals. The project uses a provided dataset and does not collect new data.
- 3. The project team has access to the data. In a real-world application, data access would be a critical privacy consideration.
- 4. The project report does not specify any data protection measures beyond standard practices for a university project.

MITIGATION

In a production environment, all data should be anonymized where possible (e.g., by reducing image resolution to a level that obscures personal details).

SUSTAINABILITY

RISK

- What is the carbon footprint of the solution?
- What types of resources does it consume (e.g. water) -and produce (e.g. waste)?
- What type of human labor is involved?
- Answer:
- 1.Training deep learning models like Segformer is computationally intensive, requiring significant electricity and contributing to a carbon footprint.
- 2.The primary resource consumed is the electricity needed to power the computers for training the models.
- 3.The project involved the labor of two students for development, experimentation, and analysis.

MITIGATION

To reduce the environmental impact, one could use pre-trained models and fine-tune them (as was done with Segformer), which is less energy-intensive than training from scratch.

Digital Ethics Canvas 2024 - C. Hardebolle, V. Macko, V. Ramachandran, Y. Do, N. Bartłomiejczyk, A. Holzer, P. Jermann