# Slide 1 - cover

Assalamu alaikum everyone I am Arik Md Isthiaque, id 1103096. Today I am going to present my thesis which title is “ArNet ArNet A Deep Learning Architecture for Pixel wise Semantic Segmentation of Images”. Arifa Islam Champa Mam is my thesis supervisor. I am very grateful to her for her support and words of encouragement.

# Slide 2 – introduction

At first, we need to know what is segmentation and semantic segmentation. Segmentation is a process that classifies the object in an image. In segmentation, the only job is to identify the object that is different from others. But in semantic segmentation, the job is to identify the same object in the same class. In the picture, you can see an example of semantic image segmentation. if you look carefully you can see that the road on both sides is recognized as the same class. But it was only segmentation It would recognize them as a different class.

# Slide 3 – challenges

The main challenge in our system is to build a deep convolutional neural network. Minimize its layer so that the processing time could be minimized. Also, try to add more classes by calculating their weights.

# Slide 4 - motivation

My thesis work is motivated by SegNet architecture, which is built for understanding the road scenes. It can model shapes and appearances and understand the spatial resolution between the classes. It mostly used for autonomous driving vehicles.

# Slide 5 – literature

In the paper “feed-forward semantic segmentation with zoom-out features, a method had shown to represent the statistical structure of an image by a purely feed-forward network by mapping the Superpixels using zoom out features.

In the paper “convolutional feature masking for joint object and stuff segmentation”, a method had shown to exploit the shape information present in an image by a Convolutional neural network.

# Slide 6 – methodology

Our ArNet starts with taking RGB image as input, then the image is passed to the encoder layer. It extracts the class information from the image then is fed into the decoder layer. The decoder layer maps the class information in the feature maps and gives us classified feature maps as output.

# Slide 7 – encoder layer

In the encoder layer, we have 4 blocks. Every block is consisting of a 2D convolution operation, a batch normalization operation, a rectified linear unit activation function, and a 2D MaxPooling operation. In the last block, there is no 2D MaxPooling operation, because from here the output will be fed into the decoder layer.

# Slide 8 – encoder layer

Here the batch normalization operation is to normalize the output after each convolution operation by re-centering and re-scaling so that the performance could be improved by increasing randomness.

# Slide 9 – encoder layer

ReLU is an activation function that means a rectified linear unit. It returns 1 if the block is working well and the output is going to pass in the next block.

# Slide 10 – 2D convolution

The mathematics convolution means a combination of two functions to produce a new function. Here in image processing, the convolution is the process of estimating a set pixel values into a single pixel by multiplying them with a random set of pixels which is called kernel filters, and make a sum for the resulting pixel. In the image, the middle set pixels are the kernel filter or the kernel weight.

# Slide 11 – MaxPooling

It is a simple operation. Tt just takes the maximum pixel value from a set of pixels to represent them as a single pixel. In the image, you can see that it took 8 and 5 because they are maximum value among the 4 pixels.

# Slide 12 – decoder layer

The decoder layer is similar to the encoder layer and it also has 4 blocks. Every block is consisting of a 2D deconvolution operation, a batch normalization operation, and a 2D UnPooling operation. In the last block, there is no UnPooling operation as It is the final form and for that here we have a SoftMax activation function.

# Slide 13 – decoder layer

The 2D deconvolution operation is exactly the opposite of the 2D convolution operation. It calculates all the pixels by multiplying with the kernel weight.

# Slide 14 – decoder layer

The 2d UnPooling is exactly the opposite of 2d MaxPooling. It feels the remaining pixel or elements with zero.

# Slide 15– decoder layer

The SoftMax activation function scale the output class weights in the form of 0 to 1.

# Slide 16 – dataset

We are using the CamVid dataset, which includes precalculated class weights for our use. This dataset mainly consists of road scene images. Here you can see some random images from the dataset.

# Slide 17 – result

On the evaluation of our architecture, we got 96.3% accuracy in training and 88.7% accuracy in validation. The plotted result is shown in the image.

# Slide 18 – result

Here is a visualization of the result of our ArNet architecture. On the left you can see the input image and, in the right, you can see the semantic segmented image.

# Slide 19 – comparison

We make a comparison of our architecture with some of the state-of-the-art architecture. You can see that in comparison to the SegNet and the FCN8 we have much validation accuracy in shorter epoch time. But we weren’t able to beat the fully connected DenseNet100 which is a combination of 100 blocks and we know the by increasing the number of blocks more accuracy can be gained. But it needs too much time.

# Slide 20 – Limitations

The main limitation of our work is that our architecture is machine-oriented and it consumes too much time in low-end machines. As you saw in the previous slide it takes 32 minutes per epoch on average. Another disadvantage is we weren’t able to use more datasets as the limitation for the low-end machine configuration.

# Slide 21 – conclusion

Our architecture will improve the study of machine vision and robotics. It will also be introduced in the autonomous driving technology.

# Slide 22 – future work

In the future, we have the plan to reduce the number of layers of our architecture without reducing the accuracy so that we can lower processing time and we can able to use more datasets. We also have an idea of merging the canny edge detector algorithm to our work but this idea is still in the brainstorming stage.

# Slide 23/24 – references

These are the references I used in my presentation.

# Slide 25

Thank you, everyone, for your valuable time and your patience.