Applications of Deep Learning In Image Processing

Xu Zhang(0300062651), Mingfang Hu(101085230)
ELG 5378 Image Processing and Communication Project Report
School of Electrical and Computer Engineering, University of Ottawa

***Abstract*—Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. Deep convolutional nets have brought about breakthroughs in processing images and videos. In our project, we prepare to use CNNs to explore two of main applications of Deep Learning in image processing: Object Detection & Semantic Segmentation.**

Keywords—Deep Learning, Object Detection, Semantic Segmentation

I. INTRODUCTION

Deep Learning is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks. Convolutional Neural Networks(CNN) is a class of deep neural networks which were inspired by biological processes[1] in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field.

Convolutional Neural Networks can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

A ConvNet is able to successfully capture the Spatial and Temporal dependencies in an image through the application of relevant filters. The architecture performs a better fitting to the image dataset due to the reduction in the number of parameters involved and reusability of weights. In other words, the network can be trained to understand the sophistication of the image better. So, Convolutional Neural Networks can dramatically improve the state-of-the-art areas such as image processing and computer vision. [2] There are three types of layers in a CNN: convolution layer, pooling layer and fully connected layer. [3]

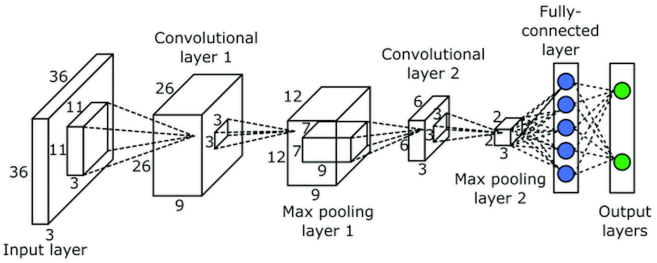


Figure 1. Architecture of Convolutional Neural Network.

A. Convolutional Layer

The convolutional layer is the core building block of a CNN. The layer's parameters consist of a set of learnable filters (or kernels), which have a small receptive field, but extend through the full depth of the input volume. During the forward pass, each filter is convolved across the width and height of the input volume, computing the dot product between the entries of the filter and the input. Let I(x,y) be an image and let f(x,y) be the filter, the convolution is

$$(I * f)(x,y) \equiv \sum_{-\infty}^{\infty} \sum_{-\infty}^{\infty} I(x-u, y-v)\, f(u,v)$$

The objective of the Convolution Operation is to extract the high-level features such as edges, from the input image. And producing a 2-dimensional activation map of that filter.

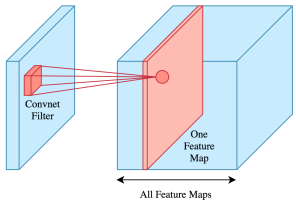


Figure 2. Convolutional Layer.

There are two types of results to the operation — one in which the convolved feature is reduced in dimensionality as compared to the input, this is done by applying Valid Padding; and the other in which the dimensionality is either increased or remains the same, this is done by applying Same Padding.

As a result, the network learns filters that activate when it extracts some specific type of feature at some spatial position in the input. ConvNets need not be limited to only one Convolutional Layer. Conventionally, the first ConvLayer is responsible for capturing the Low-Level features such as edges, color, gradient orientation, etc. With added layers, the architecture adapts to the High-Level features as well, giving us a network which has the wholesome understanding of images in the dataset, similar to how we would.

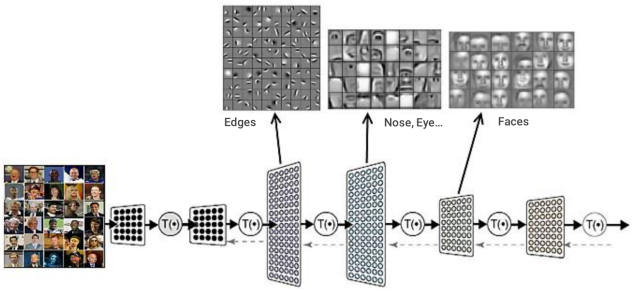


Figure 3. Feature Extraction of Convolutional Layer.

B. *Pooling Layer*

Another important layer of CNNs is pooling layer, which is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training of the model.

There are two types of Pooling: Max Pooling and Average Pooling. Max Pooling returns the maximum value from the portion of the image covered by the Kernel. On the other hand, Average Pooling returns the average of all the values from the portion of the image covered by the Kernel.

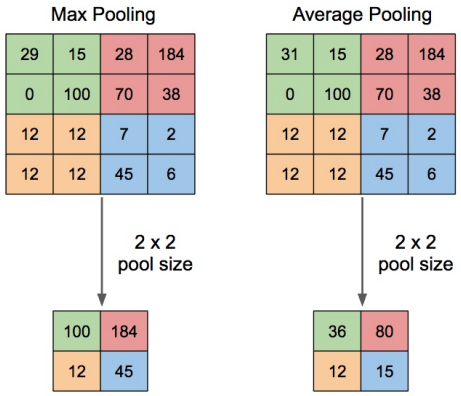


Figure 4. Comparison of Max Pooling and Average Pooling.

Max Pooling also performs as a Noise Suppressant. It discards the noisy activations altogether and also performs de-noising along with dimensionality reduction. On the other hand, Average Pooling simply performs dimensionality reduction as a noise suppressing mechanism. Hence, we can say that Max Pooling performs a lot better than Average Pooling.

The Convolutional Layer and the Pooling Layer, together form the i-th layer of a Convolutional Neural Network. Depending on the complexities in the images, the number of such layers may be increased for capturing low-levels details even further, but at the cost of more computational power.

C. *Fully Connected Layer*

Finally, after several convolutional and pooling layers, the high-level reasoning in the neural network is done via fully connected layers. Neurons in a fully connected layer have connections to all activations in the previous layer.

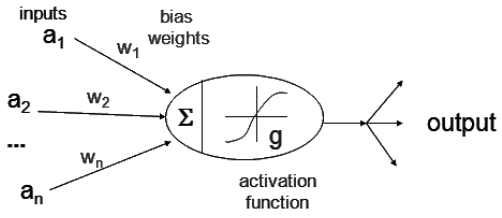


Figure 5. Single Neuron and Activation Function.

Specifically, neuron is the basic unit of computation in a neural network, often called a node or unit. It receives input from some other nodes, or from an external source and computes an output. Each input has an associated weight (w), which is assigned on the basis of its relative importance to other inputs.

To introduce non-linearity into the output of a neuron, an activation function f is applied to the output. This is important because most real world data is non linear and we want neurons to learn these non linear representations.

Every activation function (or non-linearity) takes a single number and performs a certain fixed mathematical operation on it, here are several common used activation functions:

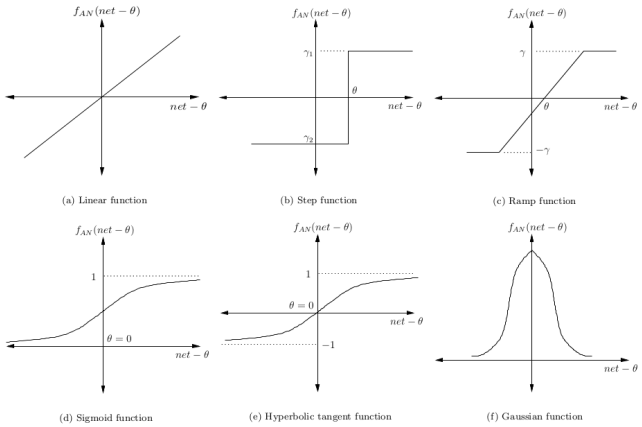


Figure 6. Some Common Used Activation Functions.

The activations of neurons can thus be computed as an affine transformation, with matrix multiplication followed by a bias offset.

We build multi-layers network through forward propagation, we arrange multiple neurons (nodes) in layers, nodes from adjacent layers have connections or edges between them. All these connections have weights associated with them and all weights in the network are randomly assigned.

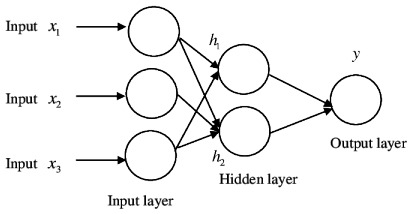


Figure 7. Forward Propagation.

Then we train our network through backpropagation. Because initially all the edge weights are randomly assigned. For every input in the training dataset, the output of multi-layers network can be calculated. This output is compared with the desired output that we already know, and the error is "propagated" back to the previous layer. This error is noted and the weights are "adjusted" accordingly. This process is repeated until the output error is below a predetermined threshold.

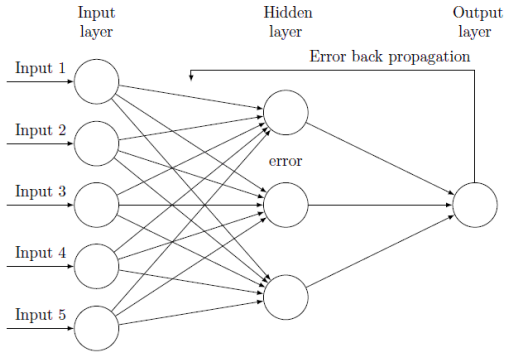


Figure 8. Backpropagation.

II. OBJECT DETECTION

As we move towards more complete image understanding, having more precise and detailed object recognition becomes crucial. In this context, one cares not only about classifying images, but also about precisely estimating estimating the class and location of objects contained within the images, a problem known as object detection.

Object detection is so popular in computer vision field, such as sports video, to judge which car is in the traffic violations.Object detection is aimed to get the information of the category and the location.We need to extract the object which we are interested in from the background.

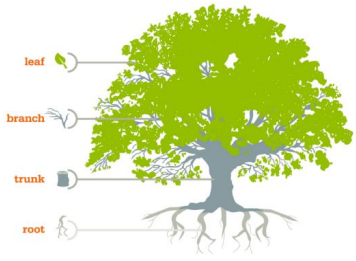
The object detection has experienced from the framework of the traditional artificial design features with the shallow classifiers to the framework of End to End object detection based on big data and deep neural networks.

A. Object Detection Using Traditional Method

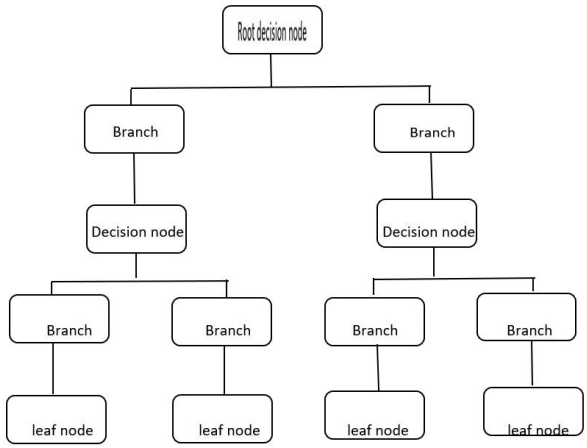
There are many ways to do the object detection, in traditional ways, the first thing is to choose the regional proposals, we scan the picture from top to bottom, and from left to right, mark every possible region, but this method needs more computation and it does not have a high accuracy and efficiency. After development, the selective search and edge box come up to reduce the computation and improve the accuracy.

The second thing is to extract the feature, such as haar, local binary patterns(LBP),histogram of oriented gradients(HOG).For example, haar are widely used because of its fast extracting speed and ability to express the variety of edge changing information. LBP are more important on expressing the texture information and it has a better adaptation to the evenly varying illumination.HOG uses the method that divide the image into small unicom area, called unit cell, then collect histogram of the edge or gradient direction of each pixel in the unit cell, finally, combine these histograms to get the feature descriptor. From the above, we could see that the traditional feature descriptor needs the researchers' experiment, as there are different requirement in different environment. The researchers need to decide how to combine the different feature descriptor to get the best efficiency.

The third thing is to choose the classifier, such as adaptive boosting(Adaboost),support vector machine(SVM) and decision tree. For example, adaboost is a iterative approach, in every step, add a weak classifier, until it could produce an enough small error probability. SVM is a supervised learning models and associated learning algorithms to analyze the data which is used to do classification and regression analysis. Decision tree is a structure of tree, and every node represents a test, every branch represents the output of the test, and every leaf node represents a category.



(a)



(b)

Figure 9. Decision Tree, (a) Intuitive Explanation, (b) Structure.

Through the above three steps, it could constitute a traditional object detection, but as described above, every method in different steps has its emphasis, so we need consider we detect the object through what kind of feature.

B. Object Detection Using Deep Learning

Deep Neural Networks exhibit major differences from traditional approaches for classification. First, they are deep architectures which have the capacity to learn more complex models than shallow ones. This expressivity and robust training algorithms allow for learning powerful object representations without the need to hand design features. This has been empirically demonstrated on the challenging ImageNet classification task across thousands of classes.

There are also many ways to detect the object in deep learning ways, such as R-CNN,FAST R-CNN and FASTER R-CNN.

(1) R-CNN:

Use selective search to get the regional proposals, and then crop the regions and resize it, and then put them into CNN to classify them , finally, the region proposals bounding boxes are refined by the support vector machine(SVM) which has been trained by the CNN.

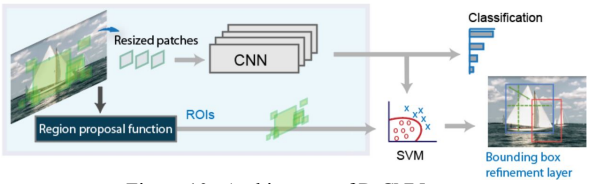


Figure 10. Architecture of R-CNN.

(2) FAST R-CNN:

Use selective search to get the regional proposals, and the difference between R-CNN with FAST R-CNN is that FAST R-CNN makes the CNN feature corresponding to each regional proposals. FAST R-CNN deal with the entire images, and it is not like R-CNN which deals with the resized regional proposals.[4]

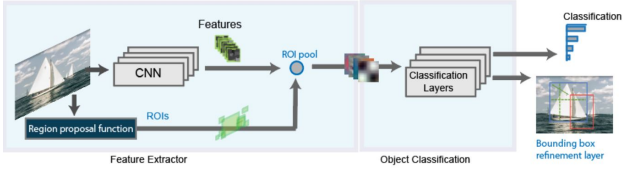


Figure 11. Architecture of fast R-CNN.

(3) FASTER R-CNN:

This method does not use the selective search to get the regional proposals, but it adds region proposal network to get the regional proposals directly.This i s the difference between the FAST R-CNN with FAST R-CNN.[5]

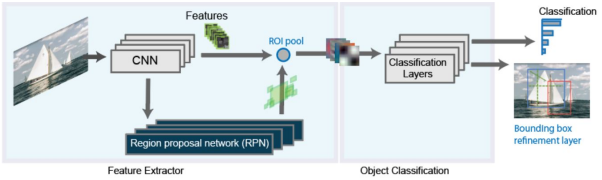


Figure 12. Architecture of faster R-CNN.

C. *Transfer Learning*

(1) R-CNN:

The last three layers are replaced by the new layers which are aimed at the object we want to detect.

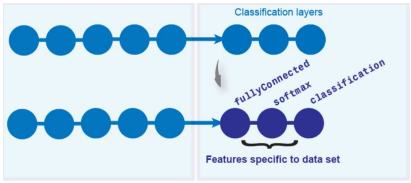


Figure 13. Transfer Learning of R-CNN.

(2) FAST R-CNN:

Based on the R-CNN, we add the ROI pooling layer(which is used to make CNN features corresponding to each regional proposal) and box regression layer(which is used to advance the object position in the image) to get the FAST R-CNN.[4]

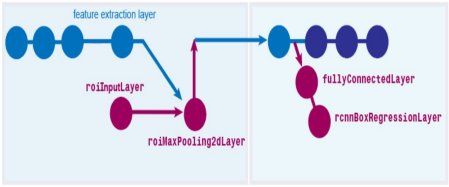


Figure 14. Transfer Learning of fast R-CNN.

(3) FASTER R-CNN:

Based on FAST R-CNN,add the RPN softmax layer to get the regional proposals directly.

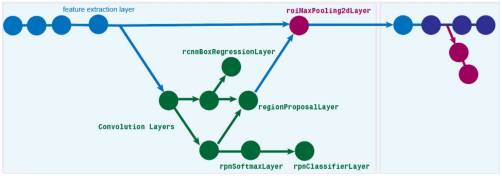


Figure 15. Transfer Learning of faster R-CNN.

D. *Implementation*

In our project, we use the R-CNN method to achieve the object detection, R-CNN is comprised by regional proposals and feature convolution in the deep neural network.[6]

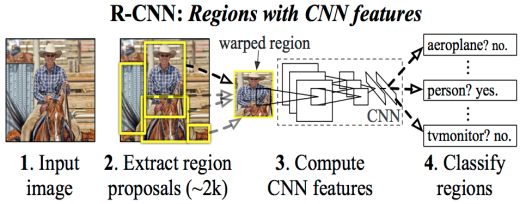


Figure 16. The Process of R-CNN.

1) Selective Search[7]

Selective search is a way to get the regional proposals. Instead of a single technique to generate possible object locations, selective search can diversify search and use a variety of complementary image partitionings to deal with as many image conditions as possible, this method results in a small set of data-driven, class-independent, high quality locations, yielding high recall. The reduced number of locations compared to an exhaustive search enables the use of stronger machine learning techniques and stronger appearance models for object recognition. In this paper we show that our selective search enables the use of the powerful Bag-of-Words model for recognition.

The first thing to perform selective search is to get the initializing region, with

R = {r1, r2, r3, r4, …., rn}

S = φ(null set)

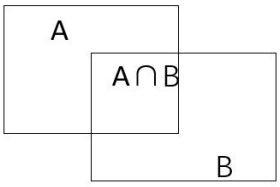
Then, we need to compute the comparability of every subset in R, and then we could get the biggest comparability called S({ri, rj}, and we would remove the ri , rj, from R.

$$rt = ri \cup rj$$

Repeat this step, until there is no element in R, we could get the real regional proposals.

2)Intersection over Union(IOU)

Objects in an Image are detected with a simple box plotted around them. The bounding box for an Object in Image is primarily hand labeled and can be called as Primary Boundary Box. The Deep Learning model predicts a bounding box around the Object which can be called Predicted Boundary Box. IOU can be computed as Area of Intersection divided over Area of Union:



A: artificial markers

B: regional proposals

IOU=A∩B/A∪B

Figure 17. IOU

This gives us an option to consider the object detected is complete or not. The IOU is a simple way of evaluation of our training model +bounding box with its performance on the testing set.

3)Non-maximum Suppression(NMS):

Non-maximum Suppression can restrict the number which is not the biggest. Through the selective search, we could get 2000 regional proposals, but some of them is not useful, so we need to use NMS to get the most accurate one.We could use the support vector machine(SVM) classifier to get the probability of belonging to the object we want to detect.We use IOU to begin from the rectangular box with the biggest probability,with the other rectangular box, if the result is bigger than the threshold value, we could remove the latter rectangular box. And we repeat this step, finally, we could get the reserved rectangular boxes.

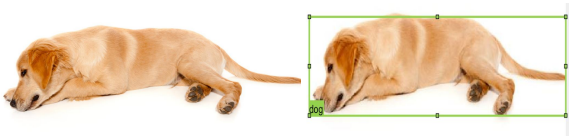
E. Regional Proposals

There are many ways to get the regional proposal, such as selective search[7] multi-scale combinatorial grouping and so on.[8] and in the experiment we choose the selective search. Through the computation of IOU and NMS, we could get the accurately regional proposals.

Figure 18. Regional Proposals.

F. Feature Convolution

In this step, we firstly manually labeled the original images to get ground-truth. After that, we combine our original images and ground-truth into training sets. Then, we can feed our training sets into our object detection model and train the model to perform object detection on new images.



(a) (b)

1 imageFilename	2 dog
'C:\Users\25172\...	[35,51,426,450]
'C:\Users\25172\...	[287,87,175,206;179,132,130,201]
'C:\Users\25172\...	[51,188,446,313]
'C:\Users\25172\...	[35,10,303,487]
'C:\Users\25172\...	[76,48,366,565;422,36,348,594]
'C:\Users\25172\...	[196,62,852,803]
'C:\Users\25172\...	[64,25,1592,3240]
'C:\Users\25172\...	[74,27,363,278]
'C:\Users\25172\...	[451,146,801,667]
'C:\Users\25172\...	[17,8,475,493]
'C:\Users\25172\...	[10,39,386,383]
'C:\Users\25172\...	[138,120,148,167]
'C:\Users\25172\...	[130,8,306,270]
'C:\Users\25172\...	*4x4 double*
'C:\Users\25172\...	[162,53,167,260]
'C:\Users\25172\...	[651,616,1121,556]
'C:\Users\25172\...	[198,23,303,253]
'C:\Users\25172\...	[172,69,433,706]
'C:\Users\25172\...	[202,111,795,556]
'C:\Users\25172\...	[519,32,457,615]
'C:\Users\25172\...	[75,28,357,231]

(c)

Figure 19. Ground Truth(Artificial Markers): (a) Original Image, (b) Labeled Image, (c) Ground Truth Table.

G. Training

In this experiment, we use the supervised pre-training, which is also called transfer learning. We use Alexnet to do the continue training. AlexNet competed in the ImageNet Large Scale Visual Recognition Challenge on September 30, 2012. The network achieved a top-5 error of 15.3%, more than 10.8 percentage points lower than that of the runner up.

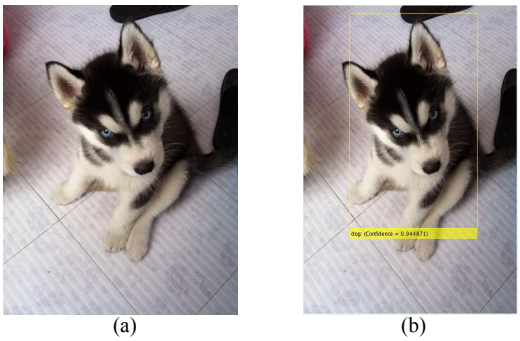
1	'data'	Image Input	227x227x3 images with 'zerocenter' normalization
2	'conv1'	Convolution	96 11x11x3 convolutions with stride [4 4] and padding [0 0 0 0]
3	'relu1'	ReLU	ReLU
4	'norm1'	Cross Channel Normalization	cross channel normalization with 5 channels per element
5	'pool1'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
6	'conv2'	Grouped Convolution	2 groups of 128 5x5x48 convolutions with stride [1 1] and padding [2 2 2 2]
7	'relu2'	ReLU	ReLU
8	'norm2'	Cross Channel Normalization	cross channel normalization with 5 channels per element
9	'pool2'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
10	'conv3'	Convolution	384 3x3x256 convolutions with stride [1 1] and padding [1 1 1 1]
11	'relu3'	ReLU	ReLU
12	'conv4'	Grouped Convolution	2 groups of 192 3x3x192 convolutions with stride [1 1] and padding [1 1 1 1]
13	'relu4'	ReLU	ReLU
14	'conv5'	Grouped Convolution	2 groups of 128 3x3x192 convolutions with stride [1 1] and padding [1 1 1 1]
15	'relu5'	ReLU	ReLU
16	'pool5'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
17	'fc6'	Fully Connected	4096 fully connected layer
18	'relu6'	ReLU	ReLU
19	'drop6'	Dropout	50% dropout
20	'fc7'	Fully Connected	4096 fully connected layer
21	'relu7'	ReLU	ReLU
22	'drop7'	Dropout	50% dropout
23	'fc8'	Fully Connected	1000 fully connected layer
24	'prob'	Softmax	softmax
25	'output'	Classification Output	crossentropyex with 'tench' and 999 other classes

Figure 20. Layers of Alexnet.

Alexnet has 25 layers, and we need to adjust the last three layers(fully connected, softmax and classification output) to do the training.Because we need to detect the dog from the background, we should set the numclasses (fully connected layers' parameter) equal to 2. We put the data of the ground truth into the CNN, and after thousands of times iteration, we could get the completely trained neural network of the dog detection.

H. Testing

We use the selective search on the tested image and use IOU and NMS to judge which region is the accurate region and keep them. Then convolve the data of the region with the trained CNN to get the final result.



(a) (b)

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Mini-batch Loss	Base Learning Rate
1	1	00:00:03	53.13%	1.6883	1.0000e-06
1	50	00:02:25	84.38%	0.3664	1.0000e-06
1	100	00:04:46	84.38%	0.4290	1.0000e-06
1	150	00:07:08	81.25%	0.6735	1.0000e-06
1	200	00:09:14	96.88%	0.1540	1.0000e-06
1	250	00:11:20	93.75%	0.3563	1.0000e-06
1	300	00:13:24	96.88%	0.0874	1.0000e-06
2	350	00:15:27	93.75%	0.1536	1.0000e-06
2	400	00:17:29	90.63%	0.1521	1.0000e-06
2	450	00:19:33	78.13%	0.4172	1.0000e-06
2	500	00:21:43	84.38%	0.5306	1.0000e-06
2	550	00:23:52	87.50%	0.4123	1.0000e-06
2	600	00:25:59	84.38%	0.5992	1.0000e-06
3	650	00:28:06	96.88%	0.3400	1.0000e-06
3	700	00:30:19	96.88%	0.1575	1.0000e-06

(c)

Figure 21. Our Results: (a) Original Image, (b) Detection Result, (c) Training Process.

As we can see above, our detection confidence can reach to 94.4871% based on 700 iterations. But there are also some drawbacks, for example, the rectangular boxing is not so accurate to mark the object, we could see this from the image, we should improve the number of the images which are inputting to the CNN to be trained and the categories of the image to make the trained CNN more accurate.

III. SEMANTIC SEGMENTATION

Semantic segmentation is one of the fundamental topics in image processing, the goal of semantic image segmentation is to label each pixel of an image with a corresponding class of what is being represented. However, we're not separating instances of the same class, we only care about the category of each pixel. As we can see in figure 19, two cows are labeled as same colour.

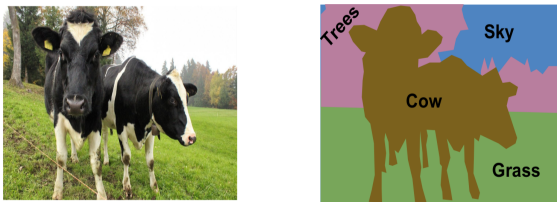
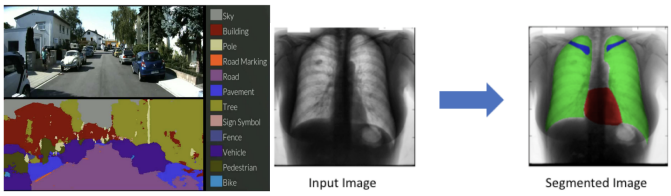


Figure 22. An example of semantic segmentation, where the goal is to predict class labels for each pixel in the image.

Segmentation models are useful for a variety of tasks, such as autonomous driving and medical image diagnostics.



(a) (b)

Figure 23. Examples of applications of semantic segmentation: (a) autonomous driving and (b) medical image diagnostics.

More specifically, our task can be represented by taking an image and using one-hot encoding[9] strategy to create output channels for each of the possible classes.

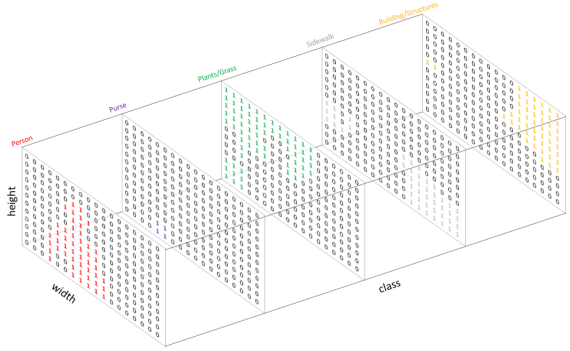


Figure 24. Output channels for each of the possible classes.

Then we overlay single channels to get segmentation map where each pixel contains a class label represented as an integer.

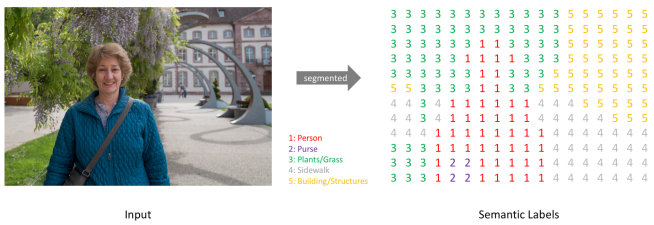


Figure 25. Segmentation Map.

One popular approach to build a computationally efficient semantic segmentation model is to build fully convolutional networks[10] with encoder/decoder structure[11], where we downsample the spatial resolution of the input, developing lower-resolution feature mappings which are learned to be highly efficient at discriminating between classes, and the upsample the feature representations into a full-resolution segmentation map.

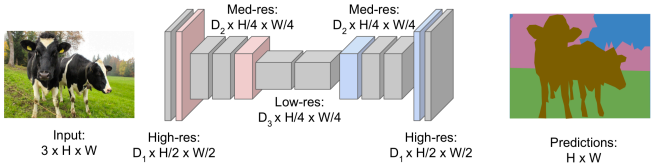


Figure 26. Architecture of semantic segmentation network.

There are a few different approaches that we can use to upsample the resolution of a feature map. Transpose convolutions are by far the most popular approach as they allow for us to develop a learned upsampling.

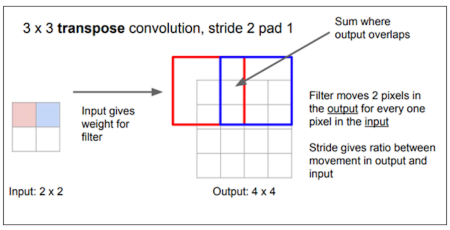


Figure 27. Transpose Convolutions.

In our project, we build our semantic segmentation model based on transfer learning technique, which is a deep learning approach in which a model that has been trained for one task is used as a starting point to train a model for similar task.

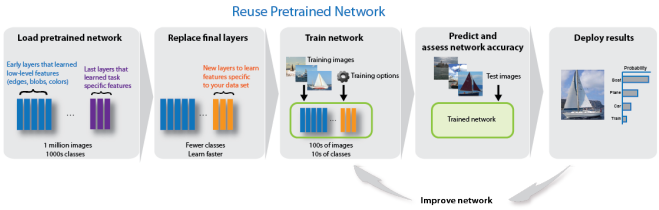


Figure 28. Workflow of Transfer Learning.

Transfer learning is commonly used in deep learning applications. You can take a pretrained network and use it as a starting point to learn a new task. Fine-tuning a network with transfer learning is usually much faster and easier than training a network with randomly initialized weights from scratch. You can quickly transfer learned features to a new task using a smaller number of training images.

The graph below shows the network performance for models with transfer learning and models trained from scratch. With transfer learning, it is possible to achieve a higher model accuracy in a shorter time.

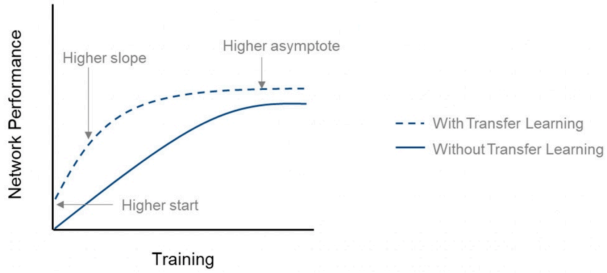


Figure 29. Network performance of training from scratch and transfer learning.

A. *Datasets*

We choose Camvid Datasets as our datasets. The Cambridge-driving Labeled Video Database (CamVid) is the first collection of videos with object class semantic labels, complete with metadata. The database provides ground truth labels that associate each pixel with one of 32 semantic classes.

However, the classes in CamVid are imbalanced, as shown in Figure 27. As we can see, the frequency of roads, buildings and sky are obviously higher than cars and pedestrians, those imbalanced classes in the training data might lead to bias. To improve training, we use class weighting to balance the classes.[12]

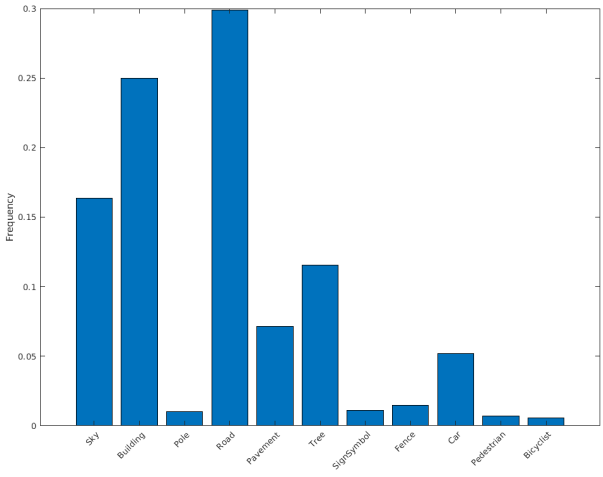


Figure 30. Classes Distribution of Camvid Datasets.

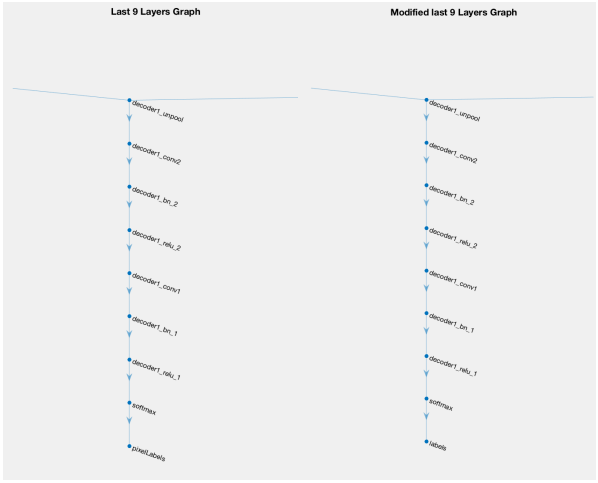
After balanced datasets, we split our datasets into training sets(60%), cross-validation sets(20%) and test sets(20%). The model is initially fit on a training dataset, that is a set of examples used to fit the parameters of the model. Successively, the fitted model is used to predict the responses for the observations in a second dataset called the validation dataset. Finally, the test dataset is a dataset used to provide an unbiased evaluation of a final model fit on the training dataset.

B. *Network*

We choose VGG-16 as our pertained network. VGG-16 is a convolutional neural network that is trained on more than a million images from the ImageNet database, the network is 16 layers deep and can classify images into 1000 object categories.

Figure 31. Description of Layers in VGG-16.

To perform transfer learning, we need to create new layers specific for our task (image sizes and the number of classes), then we remove last layer of vgg16 and add the new one we created.



(a) (b)

Figure 32. (a) Original Last 9 Layers, (b) Modified Last 9 Layers.

C. *Training*

To train our model, we select Stochastic Gradient Descent with Momentum (SGDM) algorithm as our optimization algorithm, which is an iterative method for optimizing a differentiable objective function, a stochastic approximation of gradient descent optimization. The hyper-parameters are specified as below:

Momentum = 0.9

InitialLearnRate = 1e-2

L2Regularization = 0.0005

MaxEpochs = 120

MiniBatchSize = 8

D. *Testing*

After training, we can select one picture from our results and compared it with original labeled image to quickly check our model.

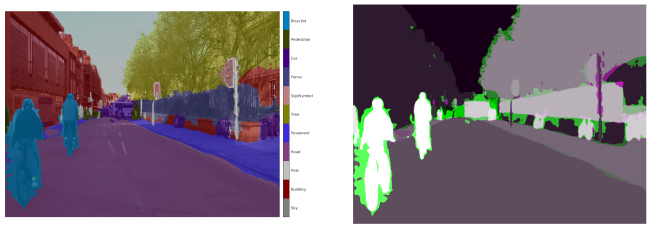


Figure 33. Result Visualization: the green and magenta regions highlight areas where the segmentation results differ from the expected ground truth.

As we can see above, the results overlap well for classes such as road, sky, and building. However, objects contain deliberate details such as pedestrians and cars are not as accurate.

Finally, to numerically measure accuracy for multiple test images, we have to use confusion matrix to evaluate our model, the results are shown below:

Classes	Accuracy
Sky	0.92695
Building	0.75324
Pole	0.70558
Road	0.92579
Pavement	0.85342
Tree	0.86732
Sign Symbol	0.76538
Fence	0.77652
Car	0.90065
Pedestrian	0.86683
Bicyclist	0.85579

Table 1. Network Performance.

E. Conclusion

Although the overall dataset performance is quite high, the class metrics show that underrepresented classes such as Pedestrian, Bicyclist, and Car are not segmented as well as classes such as Road, Sky, and Building. Additional data that includes more samples of the underrepresented classes might help improve the results.

PLAGIARISM DECLARATION

We confirm that we have read and understood the definitions of plagiarism and collusion. We confirm that we have not committed plagiarism when completing the attached piece of work, nor have we colluded with any other students in the preparation and production of this work.

REFERENCES

1. Hubel, D. H.; Wiesel, T. N. (1968-03-01). "Receptive fields and functional architecture of monkey striate cortex". *The Journal of Physiology*. **195** (1): 215–243.
2. Yann LeCun, Yoshua Bengio & Geoffrey Hinton, Deep learning, Nature volume 521, pages 436–444 (28 May 2015).
3. Ian, Yoshua and Aaron. Deep Learning, The MIT Press, November 18, 2016.
4. Girshick, Ross. "Fast r-cnn." Proceedings of the IEEE International Conference on Computer Vision. 2015.
5. Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." Advances in Neural Information Processing Systems . Vol. 28, 2015.
6. Girshick, R., J. Donahue, T. Darrell, and J. Malik. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation." Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition. Columbus, OH, June 2014, pp. 580-587.
7. J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. IJCV, 2013. 1, 2, 3, 4, 5, 9.
8. P. Arbelaez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In CVPR, 2014. 3.
9. Harris, David and Harris, Sarah. Digital design and computer architecture (2nd ed.). San Francisco, Calif.: Morgan Kaufmann. p. 129. ISBN 978-0-12-394424-5.
10. Jonathan Long, Evan Shelhamer, Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. UC Berkeley. 2014.
11. Chen, Liang-Chieh et al. “Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation.” ECCV (2018).
12. Brostow, G. J., J. Fauqueur, and R. Cipolla. "Semantic object classes in video: A high-definition ground truth database." Pattern Recognition Letters. Vol. 30, Issue 2, 2009, pp 88-97.