Computer Vision Assignment

Name:- Bishnu Kumar Kashyap

Roll No:-M20AIE227

Q-1)

C) The architecture of AlexNet contains 60,000 total parameters within 8 total layers: five convolutional layers, and three fully connected layers. Additionally, the model used ReLU (Rectified Linear Unit) activation functions, rather than tanh (hyperbolic tangent), which was standard at the time, which helped reduce the training time of the network, and was a current solution to the "vanishing gradient" problem. The pooling layers also introduce a stride (in AlexNet it was of length 4 pixels) when building the feature map, meaning that there was an overlap between each of the local receptive fields, which significantly reduced the error of their model.

Below are the implementation of AlexNet model with dataset:

Colab:-
https://colab.research.google.com/drive/1u1DOrA6MEssP_FjkFZYvtDRc66V8Gq4O?usp=sharing

We can see we are getting the good accuracy after training model on this Alexnet.

Q-2)

A) **Local Binary Pattern** (LBP) is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number.
let’s go further and see the steps of the algorithm:

Parameters: the LBPH uses 4 parameters:

Radius: the radius is used to build the circular local binary pattern and represents the radius around the central pixel. It is usually set to 1.
Neighbors: the number of sample points to build the circular local binary pattern. Keep in mind: the more sample points you include, the higher the computational cost. It is usually set to 8.
Grid X: the number of cells in the horizontal direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.
Grid Y: the number of cells in the vertical direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.

Conclusions:-

- LBPH is one of the easiest face recognition algorithms.
- It can represent local features in the images.
- It is possible to get great results (mainly in a controlled environment).
- It is robust against monotonic gray scale transformations.
- It is provided by the OpenCV library (Open Source Computer Vision Library).

Q-2) B)

Pretrained LCNN:

Light CNN (LCNN) is CNN based model which was proposed in Interspeech 2019 by STC teams and state of the art of ASVspoof2019.

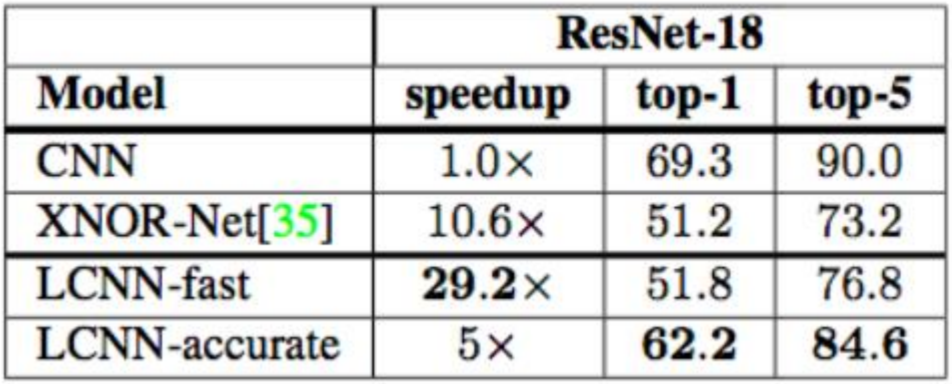
LCNN is featured by max feature mapping function (MFM). MFM is an alternative of ReLU to suppress low-activation neurons in each layer. MFM contribute to make LCNN lighter and more efficient than CNN with ReLU.

LCNN is a fast, compact, yet accurate model for convolutional neural networks that enables efficient inference and training. Training LCNN involves jointly learning a dictionary of vectors and a small set of linear combinations. LCNN is not a model architecture, but you can convert any CNN architecture to LCNN by replacing all the convolutional and dense layers to lookup-based layers. AlexNet LCNN, for example, can offer 37.6x speedup while maintaining 44.3% top-1 ImageNet accuracy, or it can acheive 55.1% top-1 ImageNet accuracy while giving 3.2x speedup.

LCNN appeared in CVPR 2017.

Inference efficiency results

Any CNN architecture can be converted to LCNN, by replacing the convolutional and dense layers to look-up based ones. Depending on the size of the dictionary and sparsity parameters, it can offer different speedup and accuracy.



	ResNet-18		
Model	**speedup**	**top-1**	**top-5**
CNN	1.0×	69.3	90.0
XNOR-Net[35]	10.6×	51.2	73.2
LCNN-fast	**29.2×**	51.8	76.8
LCNN-accurate	5×	**62.2**	**84.6**

Q-3) A

FCN — Fully Convolutional Network **(Semantic Segmentation)**

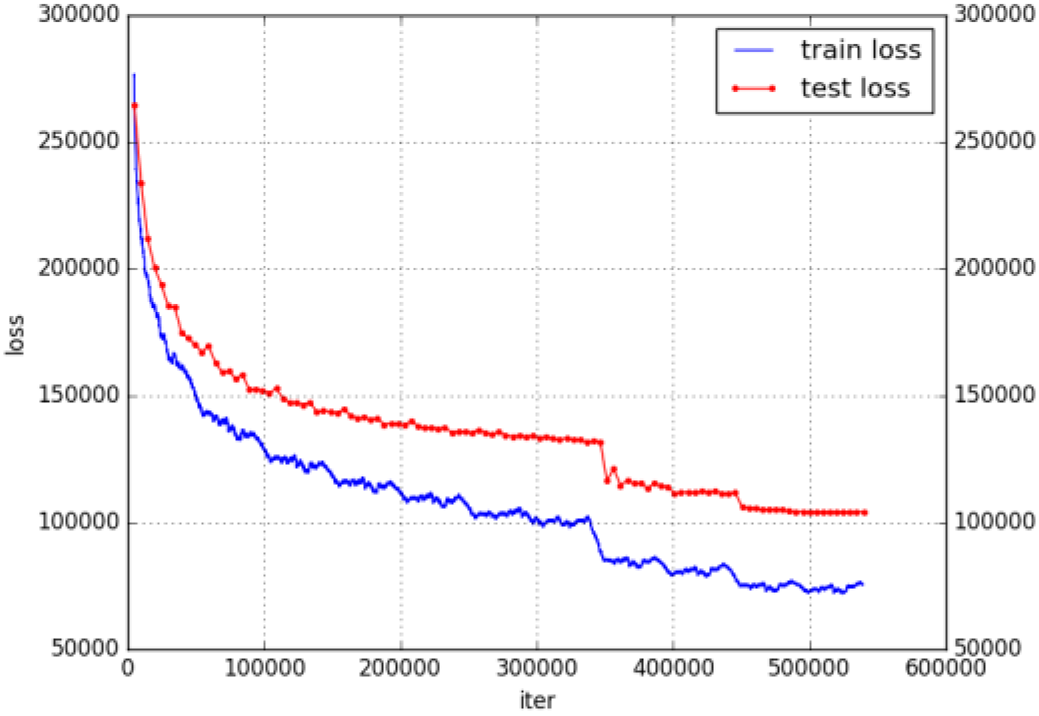
Architecture

Here compared several convolutional networks for classification and choose the VGG 16-layer net to transform it into a fully connected network. All fully connected layers of the VGG net are converted to convolutions. The final classification layer is removed and instead 1x1 convolutions are added for each class at each coarse output location. Also, a deconvolution layer is added to bilinearly gives the first fully convolutional classifier, that archievs good results on standard metrics, but still has a fairly coarse output. A common challenge with Image Segmentation is the fact that global information resolves "what" whereas local information shows "where". Convolutional Neural Networks are build in a way to go from local to increasingly global information. In order to preserve some of the local information in the coarse output layers, the authors introduce skip layers. Skip layers combine the final prediction layer with earlier layers, that have a finer stride and more local information. By doing that they transform the network from a line topology to a directed acyclic graph (DAG). Figure 2 shows how the transformation is done.

Training for the Cityscapes Dataset:

The Cityscapes Dataset is a dataset that focuses on semantic understanding of urban street scenes. It contains among other data, high-quality pixel-level annotations of 5000 frames from 50 cities with 30 different classes. The annotations are divided into three sets: training (2975 images), validation (500 images) and test (1525 images). Only the training and the validation set are public. Results from the test set are posted on the cityscapes website (http://cityscapes-dataset.com/).

Results With images downscaled by factor two, we did 550000 training iterations (one iteration corresponds to forward- and backpropagation of one image). Figure 2 shows how the loss decreases, there are visibly faster drops at about 350000 iterations and at about 450000 iterations, corresponding to the change of network architecture from FCN32S to FCN16S and from FCN16S to FCN8S.



Conclusion Converting standard classification nets to fully convolutional networks is a relatively simple, but powerful approach. We use fully-convolutional networks for semantic segmentation, both to evaluate pretrained models and to train our own models for the cityscapes dataset. We look at the problem of limited GPU memory and discuss possible solutions: downscaling, cutting images in half, multi-GPU training and training on CPUs. We archive comparable results on the cityscapes dataset. The practical helped to improve Python and Linux skills and to get familiar with the caffe framework.

Classification to Semantic Segmentation : conventionally, an input image is downsized and goes through the convolution layers and fully connected (FC) layers, and output one predicted label for the input image

Up sampling Via Deconvolution: Convolution is a process getting the output size smaller. the name, deconvolution, is coming from when we want to have up sampling to get the output size larger. And it is also called**,** up convolution, and transposed convolution**.**

Fusing the Output: After going through conv7 as below, the output size is small, then 32× up sampling is done to make the output have the same size of input image. But it also makes the output label map rough. And it is called FCN-32

SegNet:-

SegNet is a model of semantic segmentation based on Fully Comvolutional Network

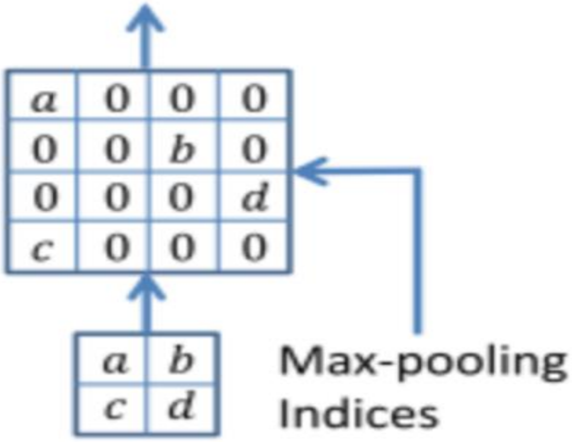
It is deep fully convolutional neural network architecture for semantic pixel-wise segmentation. This is implementation of http://arxiv.org/pdf/1511.00561v2.pdf (Except for the Up sampling layer where paper uses indices based up sampling which is not implemented in keras yet(I am working on it), but that shouldn't make a lot of difference). You can directly download the code from https://github.com/preddy5/segnet. This post is a explaination of what is happening in the code.

Architecture:

Encoder decoder architecture

Fully convolutional network

Indices pooling



a	0	0	0
0	0	b	0
0	0	0	d
c	0	0	0

a	b
c	d