CS726 Project

Foreground Segmentation for application in surveillance

FgSegNet_MyVersion

Project Statement/ Goal

In the paper titled “Foreground segmentation using convolutional neural networks for multiscale feature encoding”, authors tried to solve the problem background subtraction for various scenarios and environmental conditions.

They have modelled a deep learning based model using CNNs and used multiscale features extracted from the image to classify the pixels in image into foreground and background. As quoted by the authors “Our models are learning foreground objects from isolated frames, i.e. the time sequence is not considered during training. As a future work, we plan to redesign our network to learn from temporal data as well, and extend our FPM module.”

To elaborate the above said statement, they are using the multi-scale features of a "single image" to train the model to classify the foreground and background pixels, which may not be an entirely correct idea considering the theme of background subtraction. The background or foreground in a image depends on the past frames.

In this project, we want to try and make a deep learning model which takes the features of the past images also into consideration before classifying a pixel into foreground or background.

Generally, all classical background subtraction models should work same for similar environmental conditions. So as a preliminary study, we used the pre-trained models of the authors for inference in an “OFFICE Scenario”. The office scenario was kept almost close to the OFFICE dataset of CDNet Dataset used for training. Still, we observed that the performance was poor. This inspired us to model a network which can also take the past-information into consideration to improve the inference performance on the different scenarios, other than the dataset used for training.

Goal of this work to incorporate temporal information and model a more generic model for background subtraction

Original FgSegnet highway model was tested on the surveillance videos. But it was observed that this model does not mask off the static objects into background in a moderately different scenario indicating fitted more towards the moving objects in the dataset used for training. For example, when shown a traffic signal post in the video, the model trained on the highway dataset could not mask it off as a background object inspite of it being a static object in the video. Also when shown a single static image of text, the static text does not merge into the background. It is happening because the FgSegNet does not consider the temporal information in the videos.

We plan to modify the network to take care of this problem in the presented model.

Related Literature

Long Ang Lim, Hacer Yalim Keles, "Foreground segmentation using convolutional neural networks for multiscale feature encoding" at Pattern Recognition Letters, vol. 112, Pg. 256 - 262, 2018,issn: 0167-8655

Approaches tried

We modelled and trained two different deep learning architectures, one using 3D convolutional network and second one using CNN-LSTM combination. Both architectures have the capability of taking video input data (num of frames x channels x height x width). Detailed below:

Using 3D convolutional neural networks

In this model, video is inputted to the network directly instead of images. Input to the network is Video which is a 4D tensor: (num of frames x channels x height x width). The output of the network is a background mask of dimensions: 1 x height x width. Background mask is a binary image with ones representing the foreground and zeros representing the background objects.

Loss function used is: Binary Cross Entropy Optimiser: RMSProp

Using CNN-LSTM combination

Another approach we tried is stacking both CNN and LSTM layers. CNN layers are used to extract features from the images. Once features of sufficient length of frames are obtained, the sequence is passed into LSTM layer. This combination extracts spatio-temporal features. This kind of architecture is mostly used for video classification applications. we tried to use the LSTM output features to re-build the mask. In this we seperated the CNN and LSTM part. We used simple VGG16 network pretrained on imagenet dataset to extract the features. FC6 layer output (1x4096) are saved for every image as feature. This kind of process is very successfully applied for video classification. But in our project, we are supposed to reconstruct the mask of image size.

Faced two problems:

1. the reconstruction from a small number of features was not very successful. we dropped the architecture.
2. large model size; unable to fit into the resources we have.

Loss function used is: Binary Cross Entropy Optimiser: RMSProp

Experiments

Code

Structure of Code

Language: Python with Keras framework Environment: Linux
We used the same structure of the code as the original author. We modified all the functions to suit our application. only the structure and some trivial parts of the code, we borrowed from the original author. Thanks to Long Ang Lim for his github page.

URL: https://github.com/rajskar/CS726

Majorly code consits of 3 .py files.

1. first file containing all the required modules to build the model.
2. second file contains the data preparation and training model
3. third file containts the testing code. For testing, we can pass video from both the storage device and also the webcam. For simplicity, we made two files for both these purposes.

Details of Code

Data Preparation

We downloaded the Highway imageset of the CDNet Dataset. It contains 1700 images along with their groundtruth. We divided these images into chunks of 5 images to form a 340 number of 4D arrays. Same is applied to the groundtruth. But it is observed that the images of the dataset are captured at very high frame rate. So if 5 consecutive frames are considered, we dont observer much of a change. so we made chunks by downsampling a fraction of 3 (skipped frames while making chunks).

Also the number of zeros in any given image will be large than the number of ones. (background is more than foreground). So to handle this class imbalance, we calculate class weigths and feed them into model for better training.

Most importantly, for multiscale features, we adopted the method used in original paper. Pyramid Scaling is used to create two spatially downsampled images of h/2, w/2 and h/4, w/4 dimensions. Stack of orignal image (h,w) and scaled down images (h/2,w/2) (h/4,w/4) are concatenated and passed into the model.

Model

we divide model into two parts, encoder and decoder. Encoder is a chain of 3D convolutional layers and decoder is chain of 3D Transposed convolution layers. We borrowed style of VGG16 for Encoder part architecture, but our architecture is made of 3D Convolution layers, unlike 2D in VGG16. These concatenated multiscale images as described above are passed to the encoder to derive features. these are passed through decoder to reconstruct the binary mask of size(h,w,1). Model flow can be found in the link: https://github.com/rajskar/CS726/blob/master/model_plot_MyTrain.png

Experimental Platform description

Hardware Setup:

We trained the model using Geforece GTX 1060 6GB Nvidia Graphic Card.

Duration and other details

Epochs: 10 Duration: around 3Hrs

Experimental Results along with Commentary

We downloaded a video from youtube https://github.com/rajskar/CS726/blob/master/4K%20camera%20example%20for%20Traffic%20Monitoring%20(Road)%20%5B360p%5D.mp4 This is our test case. This is selected such that it has both static and moving objects. The outputs of both the models are provided below: Link1: Original FgSegNet Output on the test video Link2: Our improved FgSegNet Output on the test video

The improved performance is clearly visible from the outputs. We have uploaded the entire code and results on github: https://github.com/rajskar/CS726

Due to size issues, we did not share the model online.

Effort

Fraction of time in different parts of project

Datapreparation: 1/4th time Modelling two architectures and solving various problems listed above: 1/2 of the time Training & testing: 1/4 th time

Challenging part

Due to limitation of resources, modelling appropriately was challenging. Initially while training, loss was becoming NaN. Normalising data, modifying dropout regularisation etc were adopted to rectify it. Data preparation suitbaly

Modified existing FgSegNet with 3D convolutional Network has improved the performance.

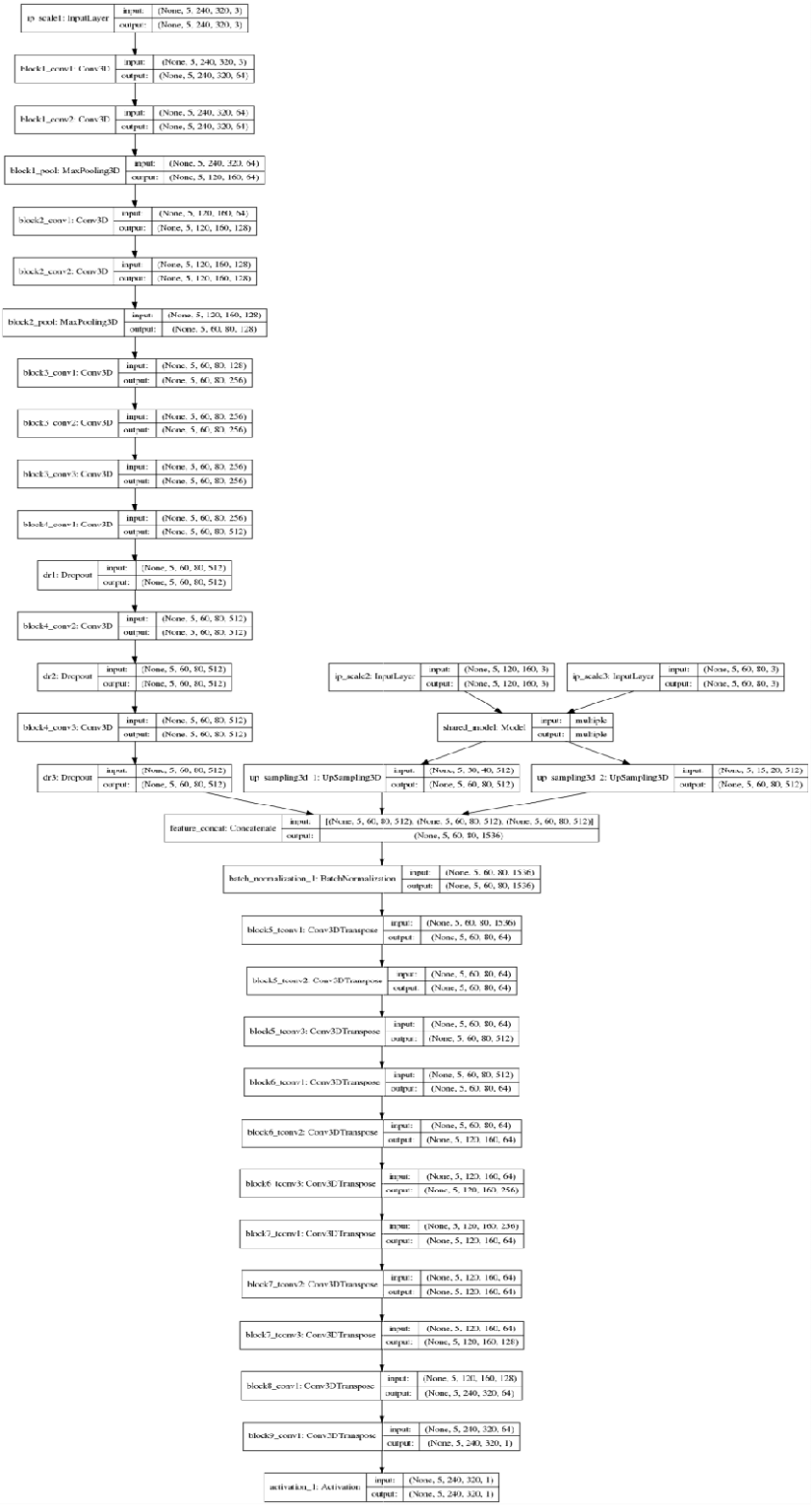

Figure 1Model Layers

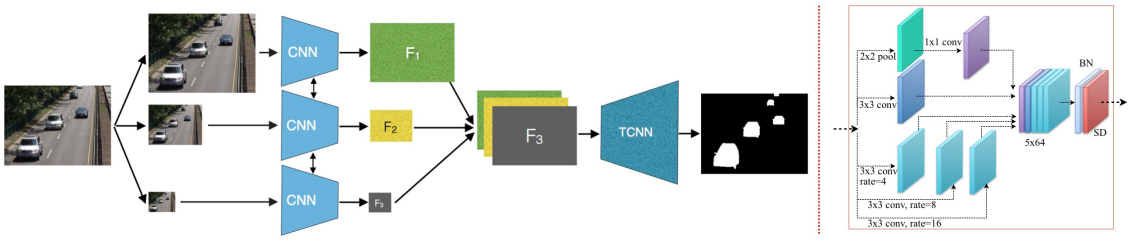

Figure 2Network Block Diagram