Assignment 2

Saurabh Kumar (2016EE10825) and Prateek Kumar Agnihotri (2016BB10033)

***Abstract*—MOOC consists of massive online material for online coaching. Because of a large content present on the internet and long lectures students find it difficult to watch the whole video as it takes a lot of time. Also most of the frames in a lecture do not contain a lot of information about the subject matter and are mostly redundant. Our work is to classify those frames which are useful in terms of information or which contain the maximum amount of information as such. We have been given a labelled data set which contains such images with labels.**

I. Data Statistics, Pre-Processing And Augmentation

Resize And Normalize

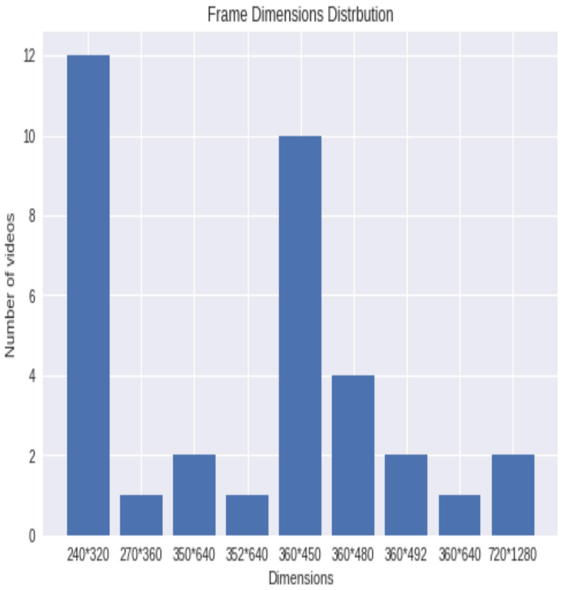


Fig. 1. Size of different frames present VS their frequency

Some videos have frames as small as 240 × 320 × 3 while others have frames as large as 720 × 1280 × 3. It is true that CNNs and RNNs can handle inputs of variable length, but such variable dimensions will be problematic while going from convolutional layers to fully connected layers. So, in order to prevent such error, all the images were resized to $240 \times 320 \times 3$ using inter-cubic interpolation.
After that as usual all frames were normalized to have zero mean and unit variance.

Data Augmentation
Boxes in the video sequence(i.e the line) are the frames which are able to able to capture all the relevant information present in the sequence. Now, during the video, prof generally does only two things- (I) Add information to the existing page/slice (II) Change the slide or page. Given above properties, one can safely assume that every box is capturing information present between previous box and the box itself.
So, In order to augment the data, we shuffled the sequence between the boxes (along with the second frame which is capturing all the relavant information in the sequence).

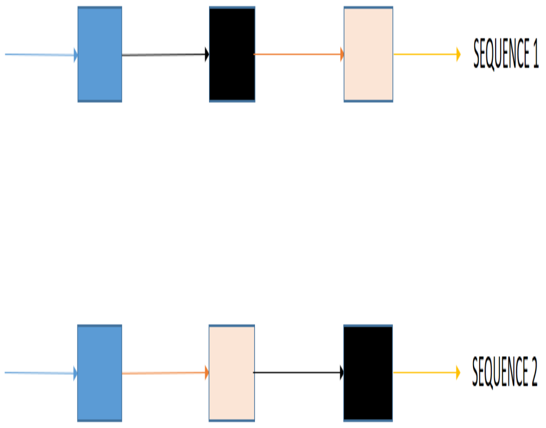


Fig. 2. Block Diagram of the Data Augmentation Model

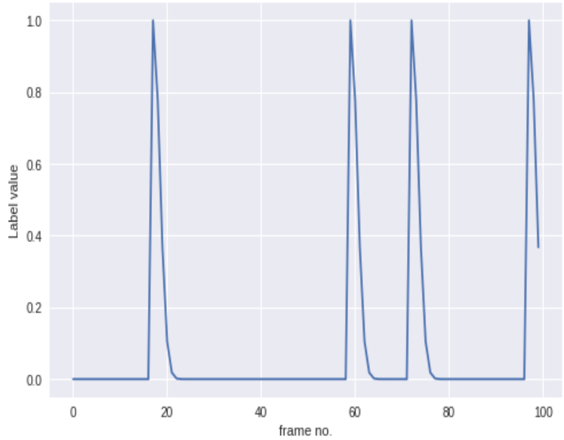
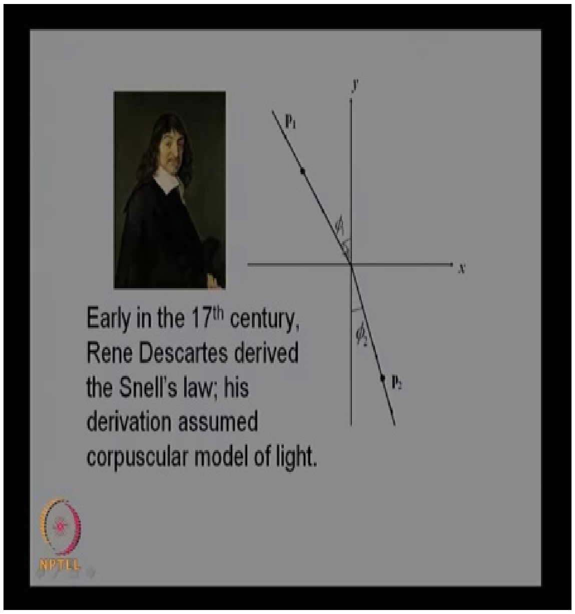


Fig. 3. Soft Label Assignment using gaussian density

Hard-Soft Label Assignment Since the the distribution of label y changes abruptly when a 1 is follwed by a large number of zeroes, we decided to normalize this effect by softening the label distribution. We call this as Hard assignment in the left of one and soft in the right. What does it help in? It helps us in various ways. Like we do in training, and instead of making the softmax layer learn 1 or 0 we make it slightly below 1, like 0.9 or 0.85, we used the same motivation to sort of ”smoothens” the curve. Also it decreases the abrupt change in label distribution hence aiding the learning process. We are not giving a mathematical proof here but just a qualitaive arguement.Also it provides a regularization, because adding noise in the labels means an L2 Regularization.

Data Visualization

The data consists of various frames which contains a snip taken from a continuous video lecture. Here we’ll try to see what these video frames contain. Some of the video frame as already mentioned above have greater dimension than the others.



Early in the 17th century, Rene Descartes derived the Snell's law; his derivation assumed corpuscular model of light.

Fig. 4. A frame taken from lecture 10

How are labels Assigned? Labels are given either a 0 or 1 depending on the relevant information they carry. For an instance an incomplete slide carries lesser information as compared to a fully completer slide, so the fully slide is assigned a label 1 and all its subsidiary slides which are incomplete but contain somewhat same information are assigned label 0. Also if there are multiple slides which have same information, that one is assigned label 1 which precedes the next uncorrelated slide in terms of information. Also as the model is sparse with respect to label 1 we expect a lot of label 0 than 1

Pixel Density Here, we define pixel density as the number of pixels in a frame which contain any texts or symbols. It is a very useful measure. Why? Because there’s a pattern that after every label 1, there is a sudden decrease in information present on the slide and henceforth this indicates the start of a new slide. A new slide means that a relevant slide has been finished and this means that we have just encountered a label 0. This gives us qualitative representation of the label in accordance to pixel density of information density. Average distance between two consecutive label 1 frames and the corresponding frequency in the whole data set. **How did we make this plot** We converted the RGB image to a gray scale image and then set a threshold of 150(on scale of 0 to 255). Any thing which is slightly grey or black will be detected as a part of text or information and will contribute to the pixel density defined above. We use python $CV2$ library for that.

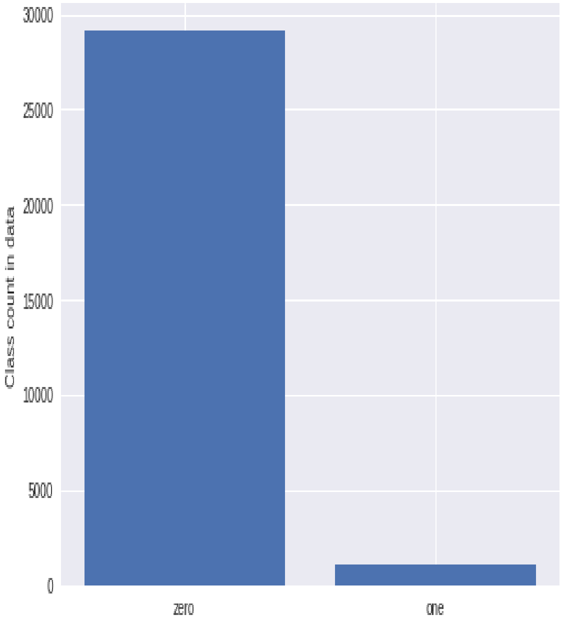


Fig. 5. Class priors

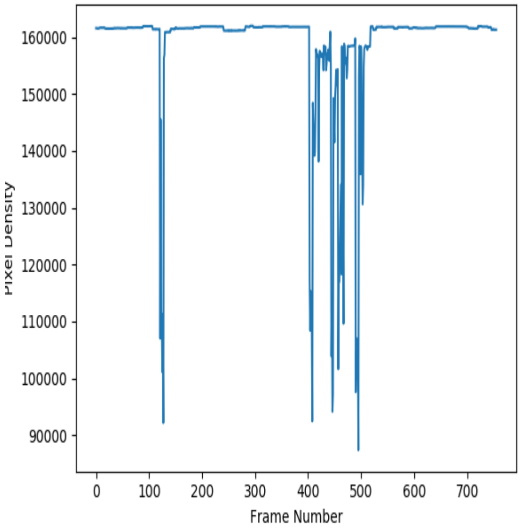


Fig. 6. Pixel Density cover of lecture 10

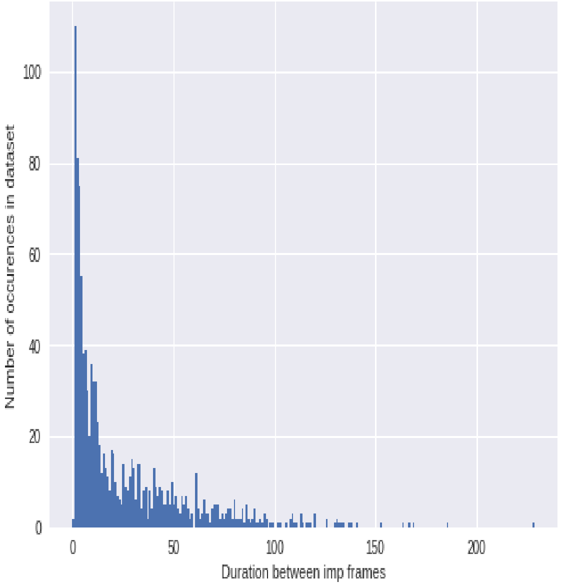


Fig. 7. Caption

II. Data Visualization using TSNE and PCA techniques

PCA and TSNE gives a very neat and concise method to represent a many fold data into a visualize able space, such as 2D or 3D by conserving maximum amount of information in a few euclidean axis. The fundamental process of PCA is to find such eigen vector which maximizes the variance of data present along that direction. Which is nothing but,

$$\sigma_{\vec{w}}^2 = \frac{1}{n} \sum_i \left(\vec{x_i} \cdot \vec{w} \right)^2 \tag{1}$$

$$\begin{aligned} &= \frac{1}{n} (\mathbf{Xw})^T (\mathbf{Xw}) \\ &= \frac{1}{n} \mathbf{w}^T \mathbf{X}^T \mathbf{X} \mathbf{w} \\ &= \mathbf{w}^T \frac{\mathbf{X}^T \mathbf{X}}{n} \mathbf{w} \\ &= \mathbf{w}^T \mathbf{V} \mathbf{w} \end{aligned} \tag{2}$$

If we differentiate this equation to maximize variance we get,

$$\begin{aligned} u &= \mathbf{w}^T \mathbf{V} \mathbf{w} - \lambda \left(\mathbf{w}^T \mathbf{w} - 1 \right) \\ \frac{\partial u}{\partial \mathbf{w}} &= 2\mathbf{V}\mathbf{w} - 2\lambda \mathbf{w} = 0 \\ \mathbf{V}\mathbf{w} &= \lambda \mathbf{w} \end{aligned} \tag{3}$$

For the case of t-SNE, the procedure is slightly complicated as t-SNE is not a linear projection of data but rather a non-linear one.

$$C = \sum_i KL \left(P_i \| Q_i \right) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}} \tag{4}$$

Where C, is the cost function involved. p and q are density functions the definition to which we omit to explain here as they require much more mathmatical reformation and assumption. But in short, this is what t-SNE does. Then we differentiate the cost function as usual to get,

$$\frac{\delta C}{\delta y_i} = 2 \sum_j \left(p_{j|i} - q_{j|i} + p_{ij} - q_{i|j} \right) \left(y_i - y_j \right) \tag{5}$$

The forms of p and q assumed are.

$$q_{ij} = \frac{\exp\left(- \left\| y_i - y_j \right\|^2 \right)}{\sum_{k \neq l} \exp\left(- \left\| y_k - y_l \right\|^2 \right)} \tag{6}$$

$$p_{ij} = \frac{\exp\left(- \left\| x_i - x_j \right\|^2 / 2\sigma^2 \right)}{\sum_{k \neq l} \exp\left(- \left\| x_k - x_l \right\|^2 / 2\sigma^2 \right)} \tag{7}$$

Now since the theory portion is complete, let us jump to the results we observed.

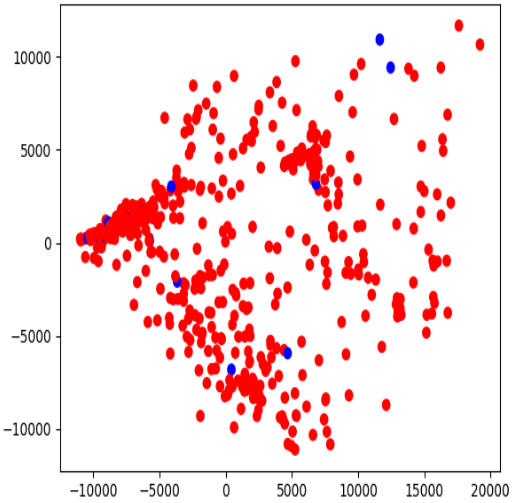


Fig. 8. 2 Dimensional PCA visualization of lecture 10

Here in this section blue dots represent label 1 and red dots represent label 0 Here as we can see the data is sparse with respect to the label 1 or the frames which have ”most information” are less in number.

Again the same in case of t-SNE plot, the data is sparse with respect to label 1.

Let us now visualize the same data but in a 3D setting. This will give additional insight about data represnetation, if any.

The data again is sparse, but we see two clusters here. These clusters maybe denote the fact that, there are a fraction of lecture slides in lecture 10 which contain images while others contain strings of equations and theory.

III. Models And Experimentation

Since we didn’t know before hand how to solve this problem or which model or technique will work better, we tried a variety of techniques to solve it. Some of which are Deep

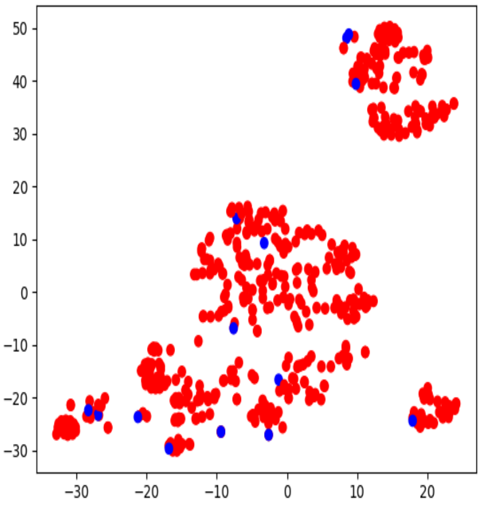


Fig. 9. 2 Dimensional t-SNE visualization of lecture 10

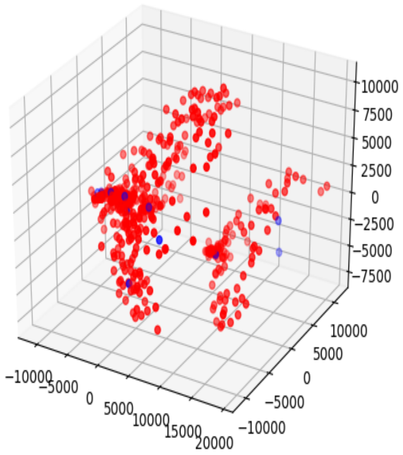


Fig. 11. 3 Dimensional t-SNE visualization of lecture 10

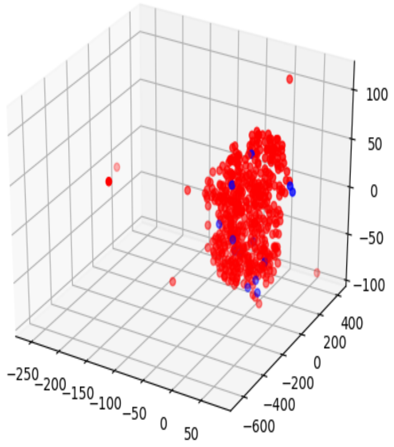


Fig. 10. 3 Dimensional PCA visualization of lecture 10

Learning based while some are not. Here we mention all our tested techniques. **The technique used are**

1) Prediction using OCR based Model
2) 3D CNN
3) Time-distributed ConvNet followed by an LSTM layer
4) Feature extraction from a Convnet(Inception Net) followed by an LSTM layer
5) 2D cnn with 3 stacked frames.

The loss function used in various Deep Learning model was binary cross entropy since our problem had 2 classes only. The corresponding loss function is,

$$J(\mathbf{w})=\frac{1}{N}\sum_{n=1}^{N}H\left(p_n,q_n\right)=-\frac{1}{N}\sum_{n=1}^{N}\left[y_n\log\hat{y}_n+\left(1-y_n\right)\log\left(1-\hat{y}_n\right)\right. \quad (8)$$

Where,

$$H(p,q)=-\sum_i p_i\log q_i=-y\log\hat{y}-(1-y)\log(1-\hat{y}) \quad (9)$$

A. Prediction using OCR based model

At first before trying any Deep Learning based model we thought this problem as a vision problem and approched it that way. We saw that the frames which have most information and which are followed by a slide which have uncorrelated and less information compared to the next frame are assigned a label 1. So we thought of extracting texts or information from the image to quantify the information available on that slide. Obviously the slide which has more information will have more number of texts or words. And then we compared the number of words or texts on different frames to assign label. We didn't expect this model to perform really well, but contrary to our surprise it worked fine. These are the results we got from our experimentation on OCR based model.

- Accuracy = 74.56%
- F1 Score = 39.2%

B. 3D CNN

3 Dimensional Convolutional Network applied for simple classification of images. The model consist of various convolutional layers followed by a fully connected neural network. 3D convolution applies the convolutions in 3D space where our third dimension is time. Here is our model. The loss function during training and validation. The blue curve represent training while the red one testing loss. We obtained

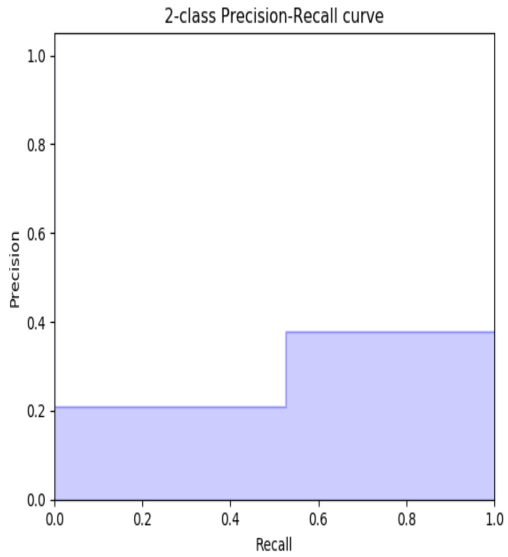


Fig. 12. ROC obtained in OCR experimentation

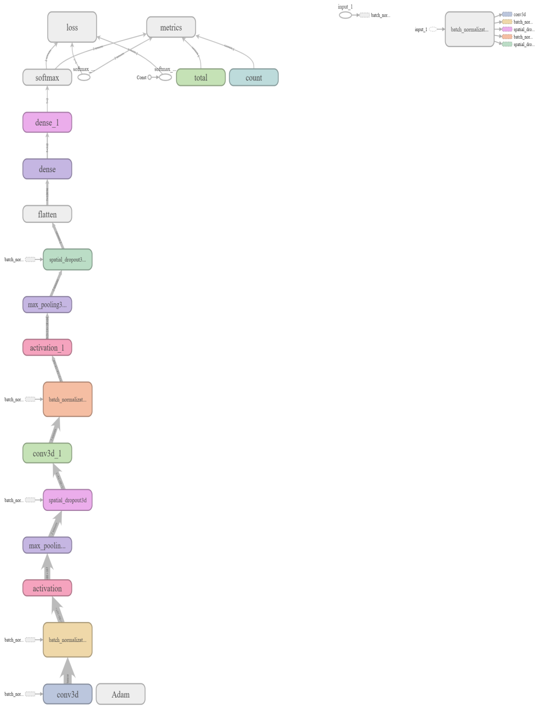


Fig. 13. Flow chart 3D CNN Graphical Model

some good results compared to OCR model which we will represent here. The results were taken during training and validation. We present the $F1$ score obtained.

C. Time-distributed ConvNet followed by an LSTM layer

Here instead of using a fourth dimension as time, we used an LTSM layer at the end of the output of the convolutional

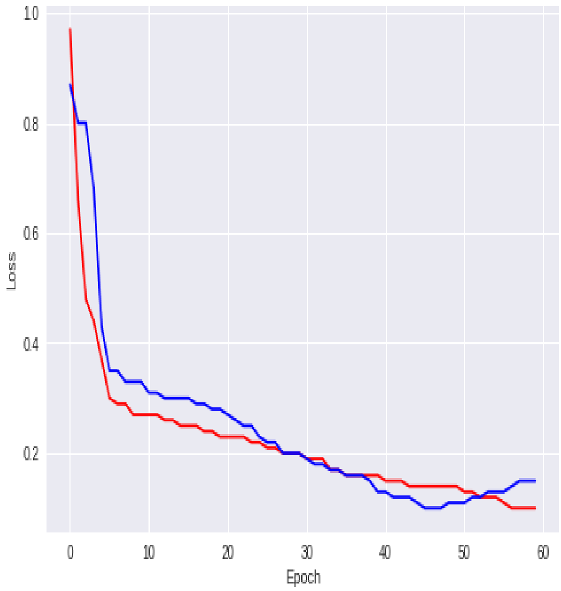


Fig. 14. Loss vs Number of iterations for 3D CNN model

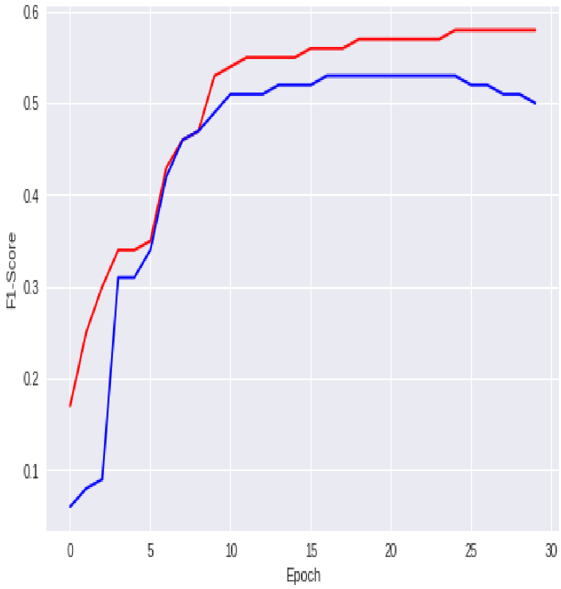


Fig. 15. F1 score curve for 3D CNN model

layer. The idea is to capture the time information along with the classification features. And using both of them together help us do that. The model used in experimentation.
The loss function during training and validation. The F1 Score obtained graph.

D. Feature Extraction from Inception Net followed by an LSTM layer

We used the trained Inception net(V3) which has been trained on ImageNet data set to extract feature from the penultimate layer of the Inception Net model. The idea is that the layers contains information about the data and it is wise

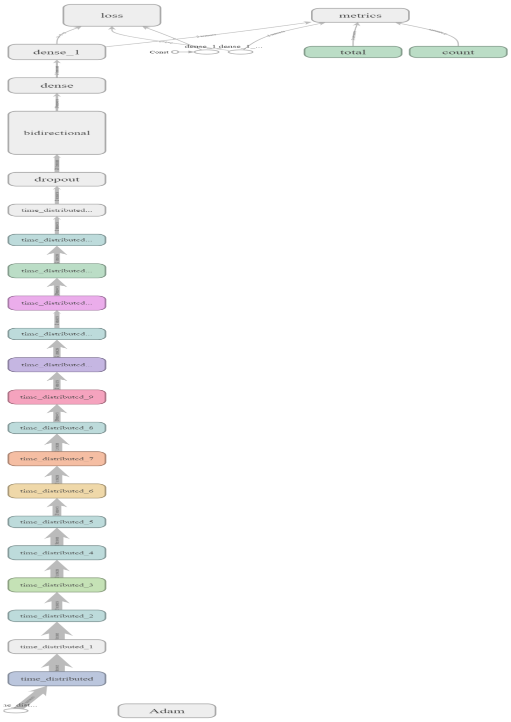


Fig. 16. Flow chart Time-distributed ConvNet Model

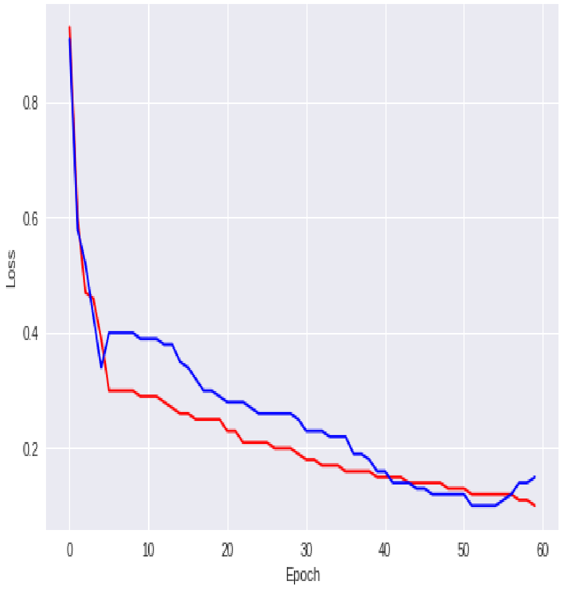


Fig. 17. Loss vs Number of iterations for Time Distributed LSTM model

to use this information rather than the information present in the real data to train our model as our real data may contain noise or redundant features. Also the training becomes faster in this case as Inception net is already trained. The loss function during training and validation. The F1 Score obtained during validation and training.

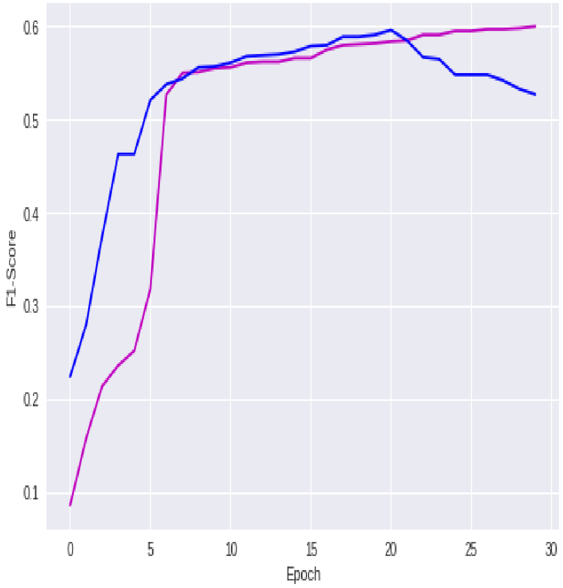


Fig. 18. F1 score curve for Time-distributed ConvNet model

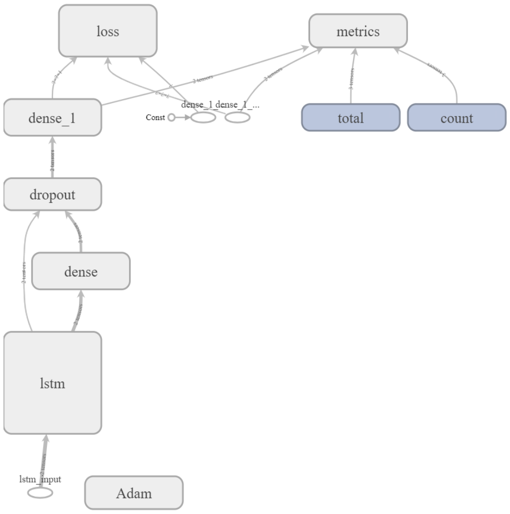


Fig. 19. Flow chart for Imagenet+LSTM model

E. 2D CNN with 3 Stacked Frames

Since only one frame is labelled as 1 among 10 or 20s or frames, it is a good idea to *maxpool* the frames itself. What we mean is, instead of training on one frame per training cycle, we stacked three consecutive frames and took that frame as a single frame and trained our CNN on that. Again the idea is that, since the information is sparsely present, the model will learn better generalization results if the information is uniformly or less sparsely present. This also provides inherent regularization because of the fact that, it reduces the noise present and also reduces the sparsity of data with label 1 with

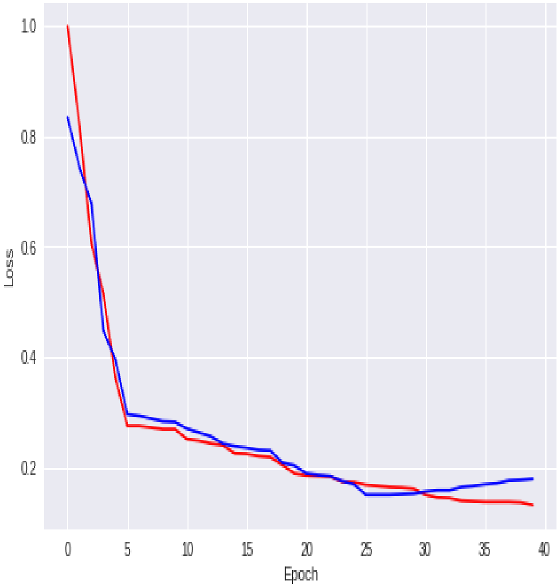


Fig. 20. Loss vs Number of iterations for Inception Net+LSTM Model

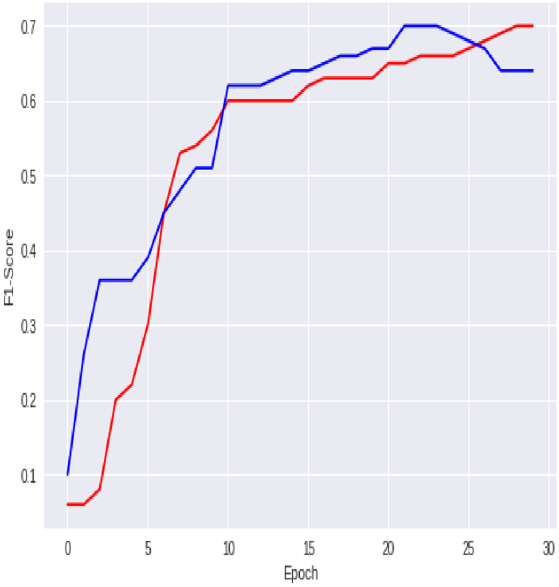


Fig. 21. F1 score curve for Inception Net + LSTM model

respect to number of training examples. Which in turn reduces model complexity.

The loss function during training and validation.

IV. Conclusion

We got to know that image classification can have varies aspects and can become a no trivial problem given the domain. As in this problem we saw how difficult it could be to classify different frames of a video. We tried a lot of models to solve the same. The data despite being sparse contains enough information for even simple vision models like OCR to solve them and hence we derived a lot of learning through this assignment. **In the process the best result we got was from the model pre-trained Inception Net followed by an LSTM layer. This can be seen from the graph of F1 scores of different models.**

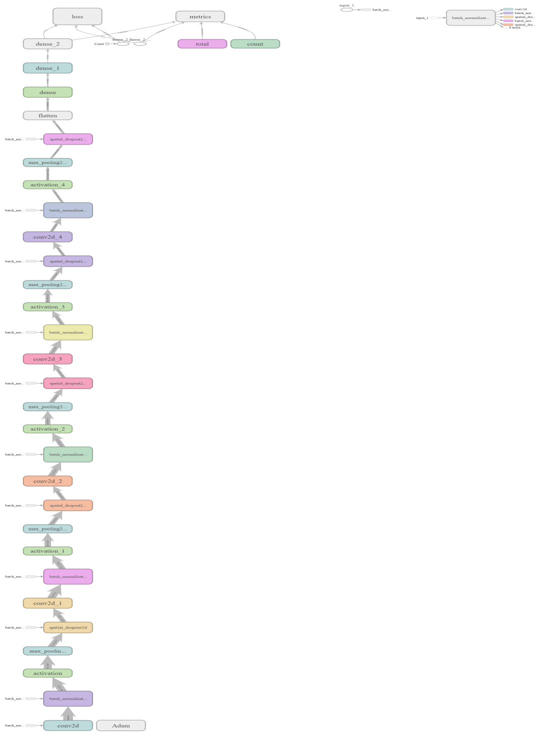


Fig. 22. Flow chart of 2D CNN with 3 Stacked Frames

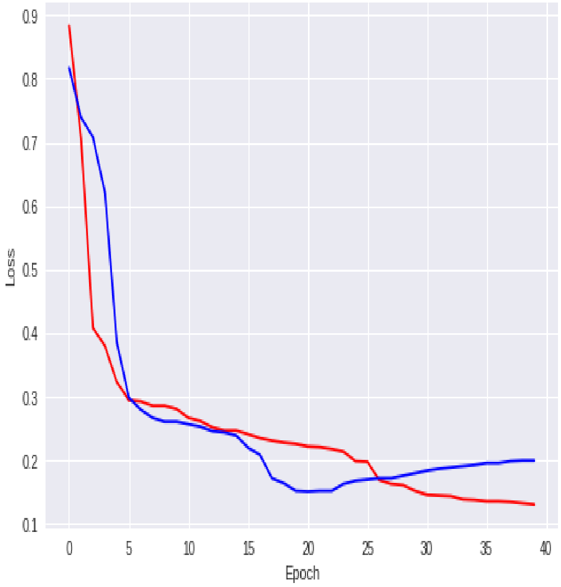


Fig. 23. Loss vs Number of iterations for 2D CNN Model