LEARNING REPRESENTATIONS FROM EEG

WITH DEEP RECURRENT-CONVOLUTIONAL NEURAL NETWORKS

Our team comprises of :

- SATYAM ANAND 2017A8PS0619P
- TANUJ KHANDELWAL 2017B1A30982P
- DEBABRATA CHAUDHURY 2017A3PS0211P

Our TA cum mentor was **Rishav Sinha**.

And thanks to our professor **Dr. Surekha Bhanot**

OUR GOAL

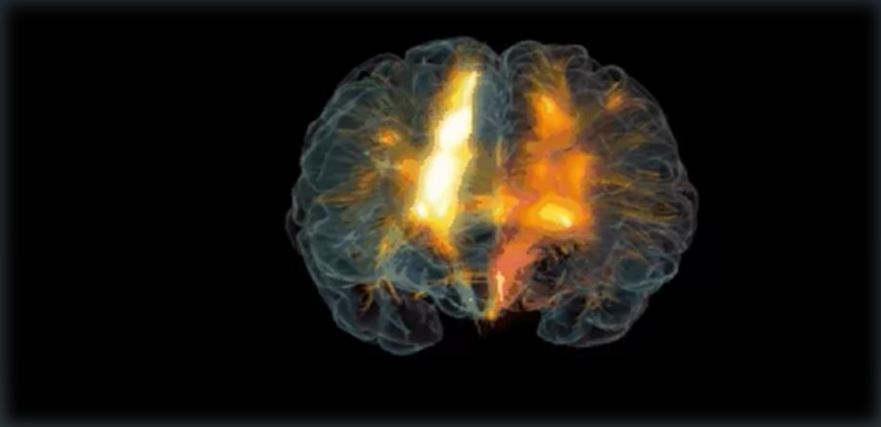
- Despite numerous successful applications of deep neural networks to large-scale image, video and text data, they remain relatively unexplored in neuroimaging domain.
- A key challenge in correctly recognizing mental states from observed brain activity is constructing a model that is robust to translation and deformation of signal in space, frequency, and time, due to inter- and intra-subject differences, as well as signal acquisition protocols.
- We propose a novel approach to learning representations from EEG data that relies on deep learning and appears to be more robust to inter- and intra-subject differences, as well as to measurement related noise.

EEG data can be thought of as a *multi-channel ”speech” signal* obtained from several ”microphones” (associated with EEG electrodes) that record signals from *multiple ”speakers” (that correspond to activity in cortical regions).*

OUR APPROACH

- Our approach is fundamentally different from the previous attempts to learn high level representations from EEG using deep neural networks.

- In other words, we obtain a sequence of topology-preserving multi-spectral images, as opposed to standard EEG analysis techniques that ignore such spatial information.

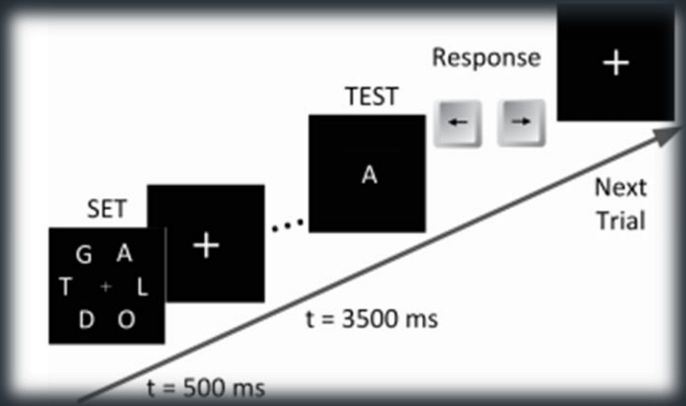


Click on Play button to play the video

- Once such EEG ”movie” is obtained, we train deep recurrent-convolutional neural network architectures, inspired by state-of-the-art video classification (Ng et al., 2015), to learn robust representations from the sequence of images, or frames.

HOW DATA IS COLLECTED

- EEG was recorded as fifteen participants (eight female) performed a standard working memory experiment from 64 electrodes placed over the scalp at standard 10-10 locations (500 Hz). Data for two of the subjects was excluded from the dataset because of excessive noise and artifacts in their recorded data.



- During the experiment, an array of English characters was shown for 0.5 second (SET) and participants were instructed to memorize the characters. Each participant was for 240 times.

- The number of characters in the SET for each trial was randomly chosen to be 2, 4, 6, or 8 with loads 1-4 respectively.
- A total of 3120 trials were recorded. Only data corresponding to correctly responded trials were included in the data set which reduced the data set size to 2670 trials.
- A number of samples equal to the test set were then randomly extracted from rest of data for validation set and the remaining samples were used as training set.

DATA PREPROCESSING

- For addressing the unbalanced ratio between number of samples and number of model parameters, as suggested we artificially expanded the dataset using data augmentation.

- We tried training the network with augmented data generated by randomly adding noise to the images. We did not use image flipping or zooming when augmenting the data due to distinct interpretation of direction and location in EEG images (corresponding to various cortical regions) as mentioned.

- We experimented with various noise levels added to each image. However, augmenting the dataset did not improve the classification performance and for higher noise values increased the error rates.

GENERATING IMAGES

- Electroencephalogram includes multiple time series corresponding to measurements across different spatial locations over the cortex.
- Fast Fourier Transform (FFT) is performed on the time series for each trial to estimate the power spectrum of the signal.
- The measurements were transformed into a 2-D image to preserve the spatial structure and use multiple color channels to represent the spectral dimension.
- Finally, we use the sequence of images derived from consecutive time windows to account for temporal evolutions in brain activity.
- We apply Clough-Tocher scheme (Alfeld, 1984) for interpolating the scattered power measurements over the scalp and for estimating the values in-between the electrodes over a 32 × 32 mesh.

3D PROJECTIONS to 2D IMAGE !!

- The EEG electrodes are distributed over the scalp in a three-dimensional space.

- In order to transform the spatially distributed activity maps as 2-D images, we need to first project the location of electrodes from a 3-dimensional space onto a 2-D surface.

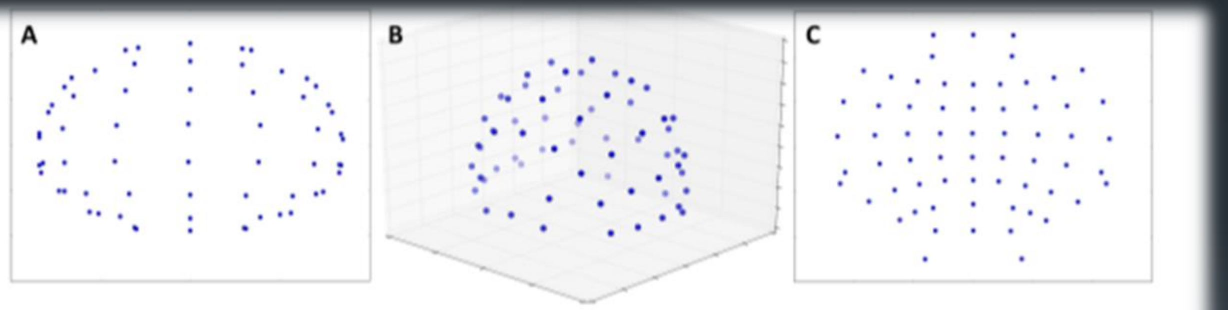


Figure 1: Topology-preserving and non-topology-preserving projections of electrode locations. A) 2-D projection of electrode locations using non-topology-preserving simple orthographic projection. B) Location of electrodes in the original 3-D space. C) 2-D projection of electrode locations using topology-preserving azimuthal equidistant projection.

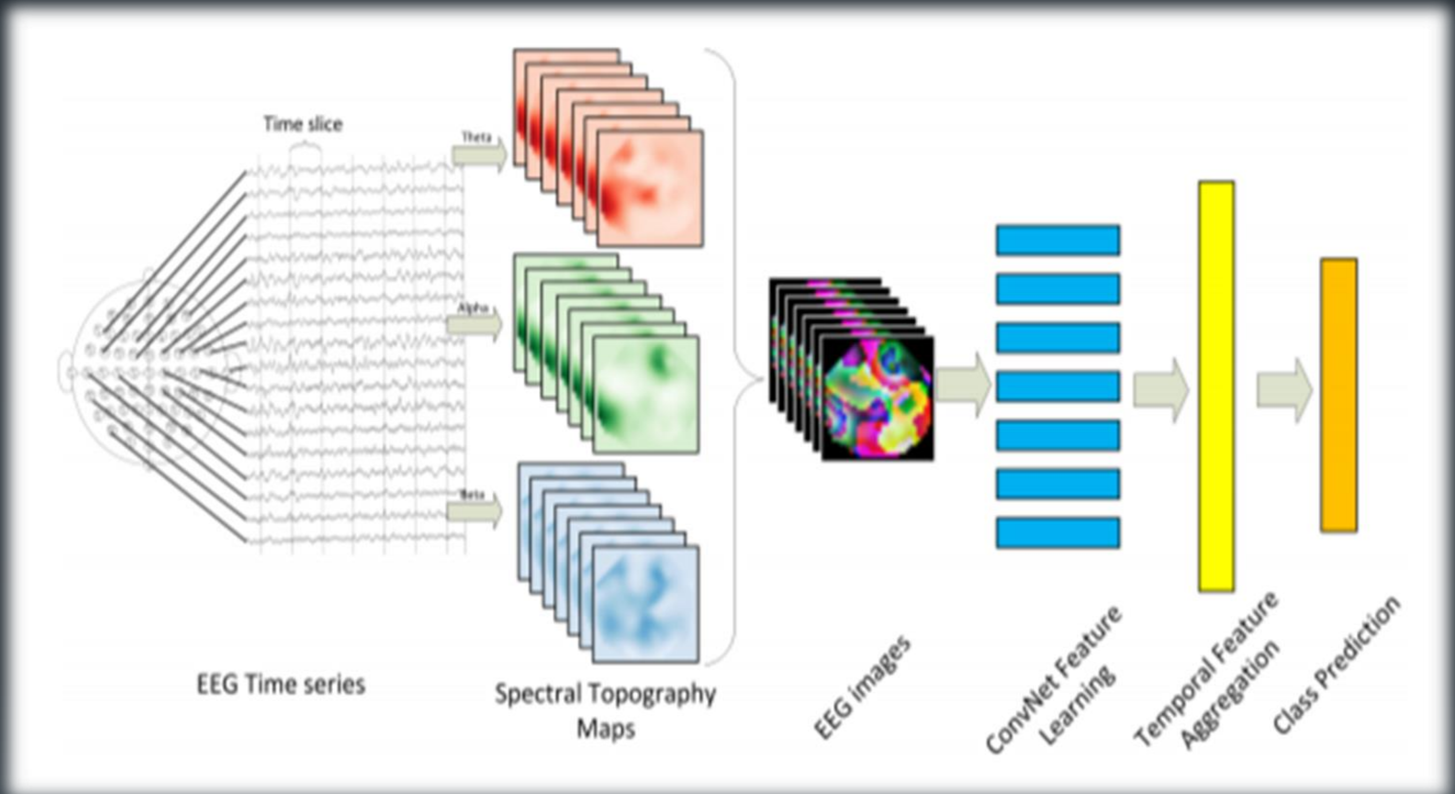
contd.

- In order to preserve relative distancing,
  - ***Azimuthal Equidistant Projection (AEP) (or Polar Projection)*** *is applied in order to get the scenario of a cap worn on the head.*
  - ***CloughTocher scheme*** *is then applied for interpolating the scattered power measurements over the scalp and for estimating the values in-between the electrodes over a 32 × 32 mesh.*

- Similarly, three topographical activity maps corresponding to each frequency band are generated. The three spatial maps are then merged together to form an image with three (color) channels.

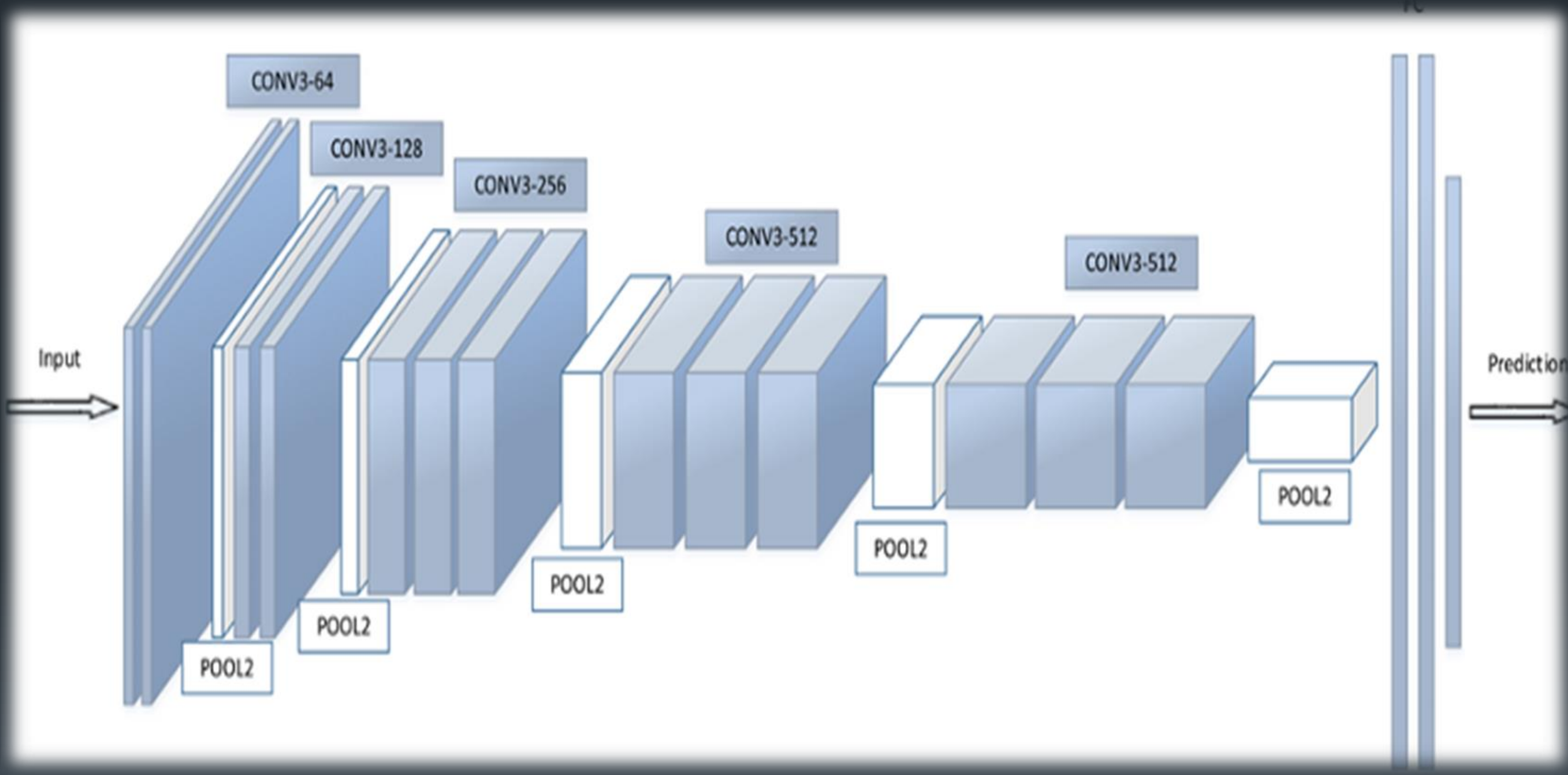
- Finally, this 3-channel image is given to our deep convolutional network.

ARCHITECTURE



(1) *EEG time series* from *multiple locations* are acquired;
(2) spectral power within three prominent frequency bands is extracted for each location and used to *form topographical maps for each time frame (image);*
(3) sequence of topographical maps are combined to form a *sequence of 3-channel images* which are fed into a *recurrent-convolutional network* for representation learning and classification.

CONVNET VGG-16 ARCHITECTURE



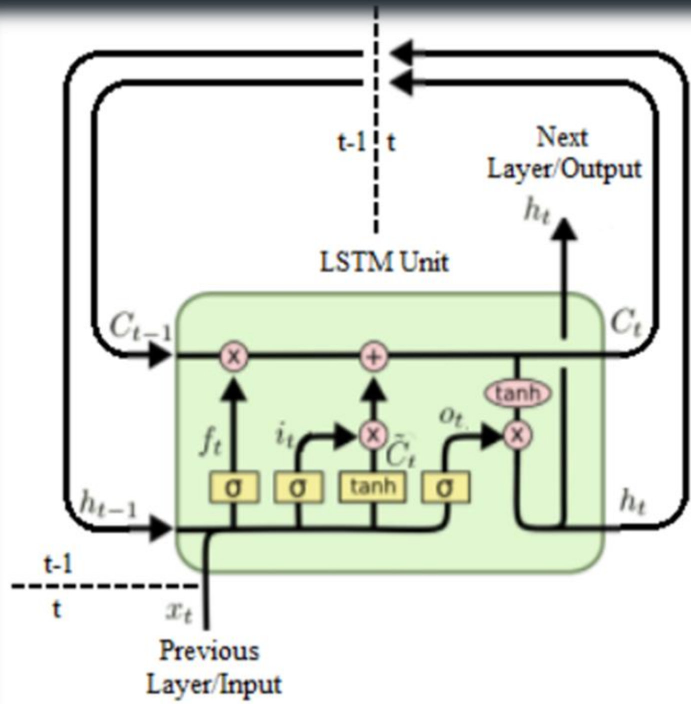
SINGLE FRAME APPROACH

ConvNet Configurations

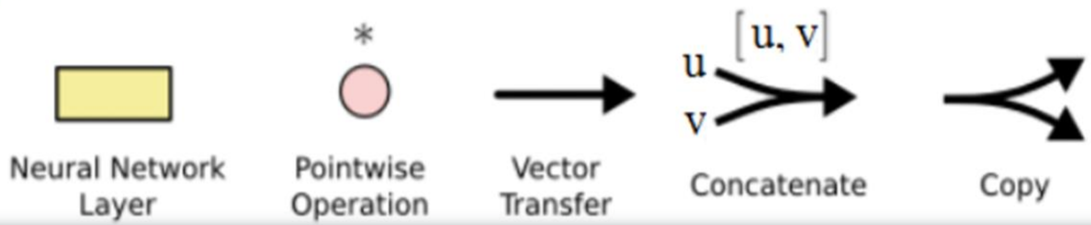
A	B	C	D
input (32 × 32 3-channel image)			
Conv3-32 Conv3-32	Conv3-32 Conv3-32	Conv3-32 Conv3-32	Conv3-32 Conv3-32 Conv3-32 Conv3-32
maxpool			
	Conv3-64 Conv3-64	Conv3-64 Conv3-64	Conv3-64 Conv3-64
	maxpool		
		Conv3-128	Conv3-128
		maxpool	
FC-512			
softmax			

The convolutional layer parameters are denoted as
conv<receptive field size>-<number of kernels>.

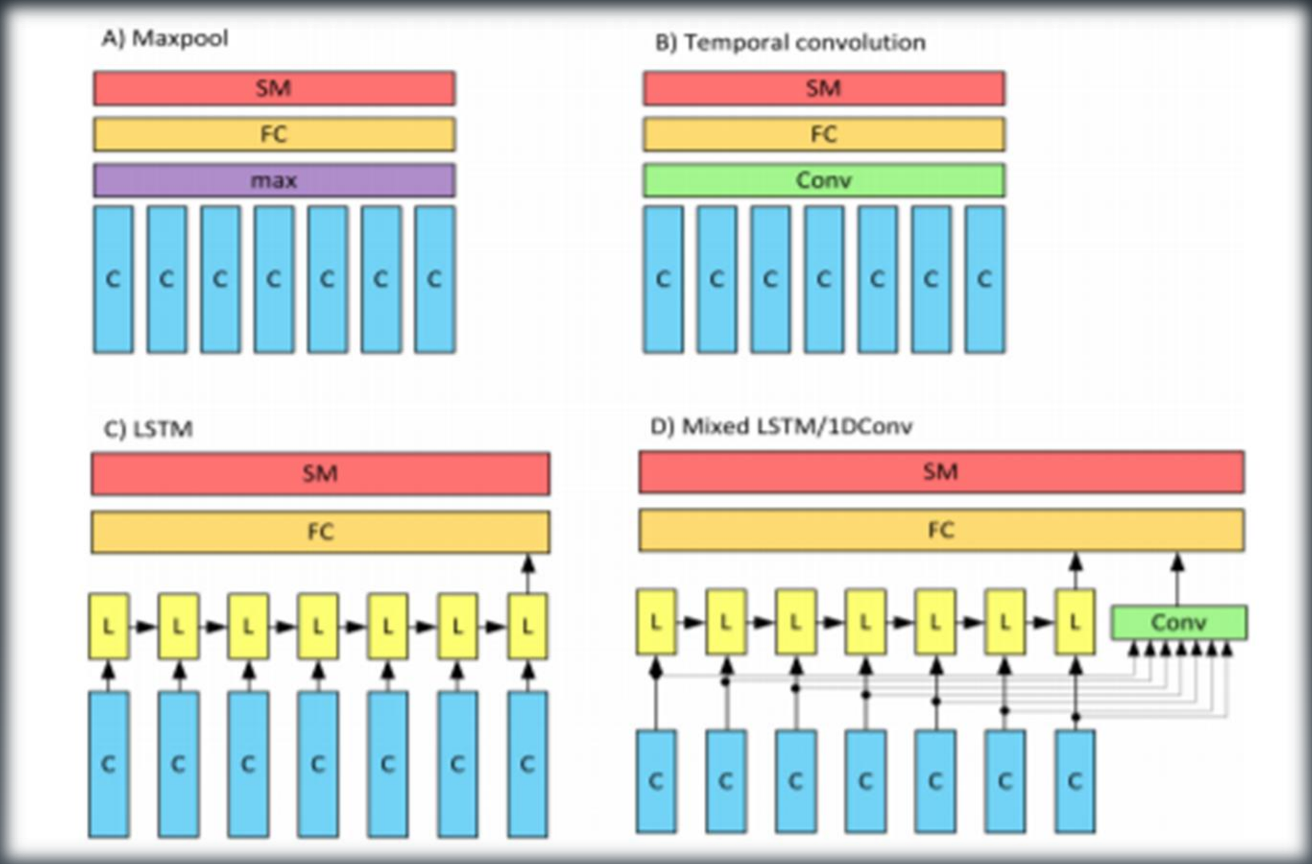
LSTM structure



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$
$$o_t = \sigma\left(W_o \ [h_{t-1}, x_t] \ + \ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$



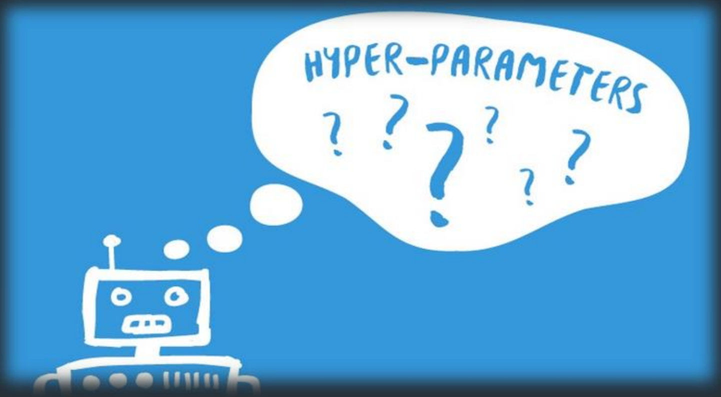
MULTI-FRAME APPROACH



TRAINING

PARAMETERS / HYPERPARAMETRS

- Each model took about 9-10 hours to train.
- Training is carried out by optimizing the cross-entropy loss function.
- Weight sharing in ConvNets results in vastly different gradients in different layers and for this reason a smaller learning rate is usually used when applying SGD.



- We trained the recurrent-convolutional network with Adam algorithm with a learning factor of 0.001. Batch size was set to 20.
- Adam has been shown to achieve competitively fast convergence rates.
- In addition, VGG architecture was used since, it requires fewer epochs to converge due to implicit regularization imposed by greater depth and smaller convolution filter sizes.

contd.

- The large number of parameters existing in our network made it susceptible to overfitting. Hence, dropout (Hinton et al., 2012) with a probability of 0.5 was used in all fully connected layers.

- Additionally, we used early stopping by monitoring model's performance over 6 Published as a conference paper at ICLR 2016 a randomly selected validation set.

- Dropout regularization has proved to be an effective method for reducing the overfitting in deep neural networks.

```
Epoch 4 of 5 took 426.157s
  training loss:              0.292986
  validation loss:            0.213847
  validation accuracy:        91.75 %
00
Final results:
  test loss:                  0.348063
  test accuracy:              89.60 %
9
01
Epoch 5 of 5 took 424.146s
  training loss:              0.274975
  validation loss:            0.185264
  validation accuracy:        91.36 %
---------------------------------------------------
Best validation accuracy:             91.75 %
Best test accuracy:           89.60 %
```

Following is a snapshot of an epoch of the second fold (or subject)

CLASSIFICATION RESULTS

TEST SUBJECT	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13
1-D Conv	64.63	88.60	86.25	98.02	98.85	97.62	97.71	97.86	97.24	95.15	90.16	82.62	52.48
LSTM	59.76	78.67	90.83	98.68	99.01	98.75	97.98	99.36	99.51	96.48	98.73	89.31	60.72
Mix	65.71	74.43	92.56	99.01	99.21	97.55	99.15	98.28	98.07	94.45	95.29	89.97	50.78

- Our classification results came very close to that mentioned in the paper apart from those of the first two folds in CNN and LSTM; and the first fold for mix-LSTM model architecture.
- This difference in our results is the result of very few training samples in the beginning due to which the model is not trained with enough data (or samples) in our case.
- Thereafter, we can clearly see that the results given by our different models is approximately same.
- Thus, we can conclude that our model performs pretty well in accordance with the results.

LEARNING OUTCOMES

- We learnt how to use *Jupyter notebook* and manage python files.
- Learning how to manage data was one of the major learning outcomes from this project.
- We had to get the data from another paper of the author and use it. We learnt how to handle data from *.mat (Microsoft Access Table)* files.
- We learnt how to apply *Azimuthal Equidistant Projection(AEP)* on our data to convert 3D points to 2D *(as can be seen in the UN logo)*.
- We learnt how to use *Google Colab* and *Github* for different purposes.


Thank
You