Epileptic Seizures Prediction Using Deep Learning Methods on EEG Recordings

G. Maragatham
Department of Information Technology, SRM Institute of Science and Technology, Kattankulathur, Chennai. E-mail: maragatg@srmist.edu.in

T. Kirthiga Devi
Department of Information Technology, SRM Institute of Science and Technology, Kattankulathur, Chennai. E-mail: kirthigt@srmist.edu.in

P. Savaridassan
Department of Information Technology, SRM Institute of Science and Technology, Kattankulathur, Chennai. E-mail: savaridp@srmist.edu.in

Sachin Garg
Department of Information Technology, SRM Institute of Science and Technology, Kattankulathur, Chennai.

Received July 24, 2020; Accepted September 30, 2020
ISSN: 1735-188X
DOI: 10.14704/WEB/V17I2/WEB17063

Abstract

Epilepsy is a neurological disorder that disturbs the brain and causes abnormal brain activity. It results in loss of awareness in some cases and unusual behavior and sensations. In this regard, if the seizures could be identified in its earlier stages then the patient can be provided appropriate care and treatment in time and prevent any severe damage to the patient as a whole. In this paper, we try to detect epilepsy using the EEG Signal Recordings and classify them using pre-trained CNN models between preictal and interictal classes. For this we are advocating the use of American Society for Epilepsy Dataset. The focus is on detecting the epilepsy pattern from the EEG recordings in a timely and accurate manner.

Keywords

Behavior or Seizures, Seizure Episodes, Nerve-signaling Chemicals.

Introduction

Epilepsy is a brain disease which is chronic noncommunicable that affects the area wherein brain activity becomes abnormal that is the central nervous system, which causes periods of unusual behavior or seizures, and are sometimes accompanied by loss of bowel control or function of bladder and also consciousness. A sudden rush of electrical activity in the brain is known as a seizure. Excessive discharge of electrical signals which happens in a group of brain cells results in seizure episodes. The sites of these discharges are in different parts of the brain. Seizures can also be caused by interruption between the normal connection that occurs in the brain between the nerve cells. This can result in high fever, a brain concussion, alcohol or drug withdrawal, high blood sugar, or low blood sugar. Epilepsy can also be caused by neurotransmitters imbalance which are the nerve-signaling chemicals.

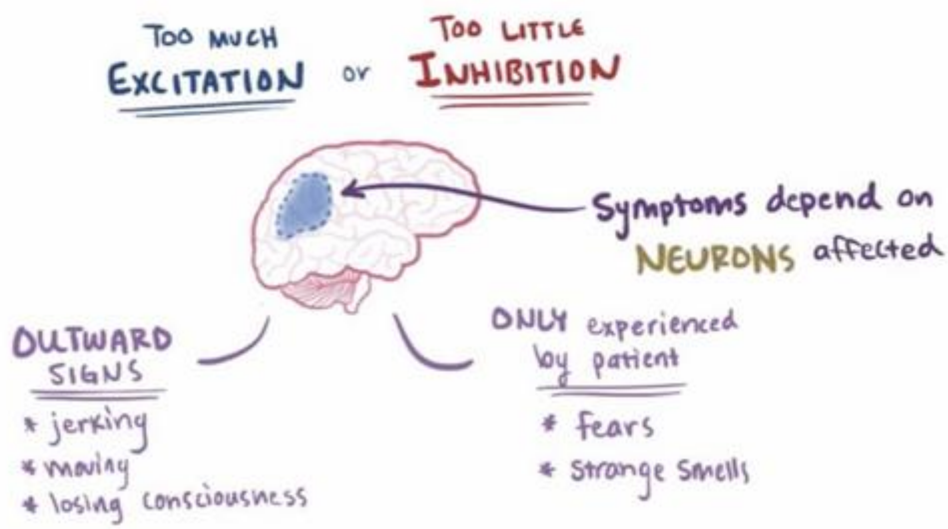


Figure 1.1 Epilepsy in brain

1) Epilepsy as a Disease

Epilepsy neurological disorder affects worldwide to around 50 million people. 4 and 10 per 1000 people is the estimated proportion of the general population at a given time with active epilepsy (i.e. treatment needed or continuing seizures). Epilepsy can occur to anyone, but it's more common in older adults and young children. It slightly occurs more in males than in females. A mild seizure may be difficult to recognize. It can last a few seconds during which you lack awareness. Seizure symptoms vary widely from briefest muscle jerks or periods of attention to intense and extended convulsions. Seizures also vary in the number of times they occur, from several a day to less than one a year. Some epilepsy patients stare blankly for a while during the occurrence of a seizure, while other patients show repeatedly twitching in their legs or arms. A single seizure doesn't necessarily mean that one has epilepsy (up to 10% of people worldwide have 1 seizure during a lifetime). Two unprovoked seizures at least are required for a diagnosis of epilepsy generally. Despite the occurrence seizures infrequently, patients having epilepsy experience tireless anxiety which is because of the possibility of seizure events occurring at any time even during carrying out essential daily activities like driving or cooking which can lead to deadly accidents.

2) Types of Seizures and its Phases

The category of seizure is decided based on what part and portion is affected in the brain, and reactions during the seizure. Seizures are categorized as Generalized (atonic, absence, tonic clonic absence) and Partial that are simple and complex seizures. These categories include various seizures types:

1. Focal or partial seizures. Caused when any side of the brain experiences abnormal electrical function. Symptoms cover hearing abnormalities and affected sense of smell. Two types of focal seizures include:

2. Generalized seizures. Both sides of the brain are included in this type which is postictal state and loss of consciousness that occur after the seizure. Generalized seizures include the following types:

There are several stages of seizures, most prominent ones include:

- **Preictal:** The time right before the occurrence of seizure. It could last from some minutes to several days and people feel and act abnormally. This stage of seizure is not experienced by not everyone. Those who do go through a preictal stage use these symptoms as a warning so they can be ready for seizure and the goal of our work is to alert the patient at this stage so that protective measures can be taken.
- **Ictal:** This stage is the period of the actual seizure. We can see actual change in a person's body physically during this stage of seizure. This is also the point where the brain of the person comes to life by the electrical storm. If the person suffering epilepsy were connected to any of the medical devices, they are sure to show metabolic, cardiovascular and changes in EEG at this point.
- **Interictal:** The time between seizures are known as the interictal stage. Maximum number of people having epilepsy, which also includes people with temporal lobe epilepsy that is more than half, suffer from emotional disruptions between these seizures. These disruptions range from fear that is mild to depression anxiety that is of pathological levels

3) Seizures Monitoring Methods

1. Electroencephalogram (EEG).
2. High-density EEG.
3. CT scan i.e Computerized tomography
4. MRI i.e Magnetic resonance imaging
5. Functional MRI (fMRI).
6. Positron emission tomography
7. Single-photon emission computerized tomography

Motivation

Patients with epilepsy can be helped by Seizure forecasting systems which lead more common and lives that are fulfilling. In order to make systems work, we need algorithms that must reliably classify the seizure occurrence by the periods of their increased probability. If the states of seizure can be accurately classified then it would be possible to put together devices which could alert the patients of the incoming seizures. Patients then will be able to avoid activities that are potentially dangerous like swimming or driving or cooking, and to prevent the incoming seizures medications could be administered when needed, reducing side effects overall that the patient's body has to bear.

1) Problem Statement

Predicting the seizure with better accuracy and simplified manner so that the prediction can be done on an embedded device or by using remote calls to a server to get the predictions. The aim is to reduce the time and complexity involved in the process of predicting while maintaining a low false positive rate. The aim is to also explore the latest computer vision models and to monitor their performance on the standard datasets available and to explore how transfer learning can be used to generalize the model's performance across different patients.

2) Approach

This solution involves the use of pretrained CNN models to classify between Interictal and Preictal stages from EEG recordings of the diseased patients. The raw EEG signals are converted into spectrograms which allows us to leverage the classification power of computer vision models. Then using this model, predictions can be made on the patient recording segments to determine the possibility of whether the segment precedes a seizure event or not.

a) The Dataset

The training data is divided into 10 minutes each of EEG-clips which are marked as "Preictal" for data sections showing state right before a seizure, or "Interictal" for data sections showing normal state. These data sections are kept sequentially, while testing ones are in random order. Within folders data sections are kept in.mat files like preictal-segment-n.mat and Interictalsegment-n.mat, for test they are kept as test-segment-N.mat.

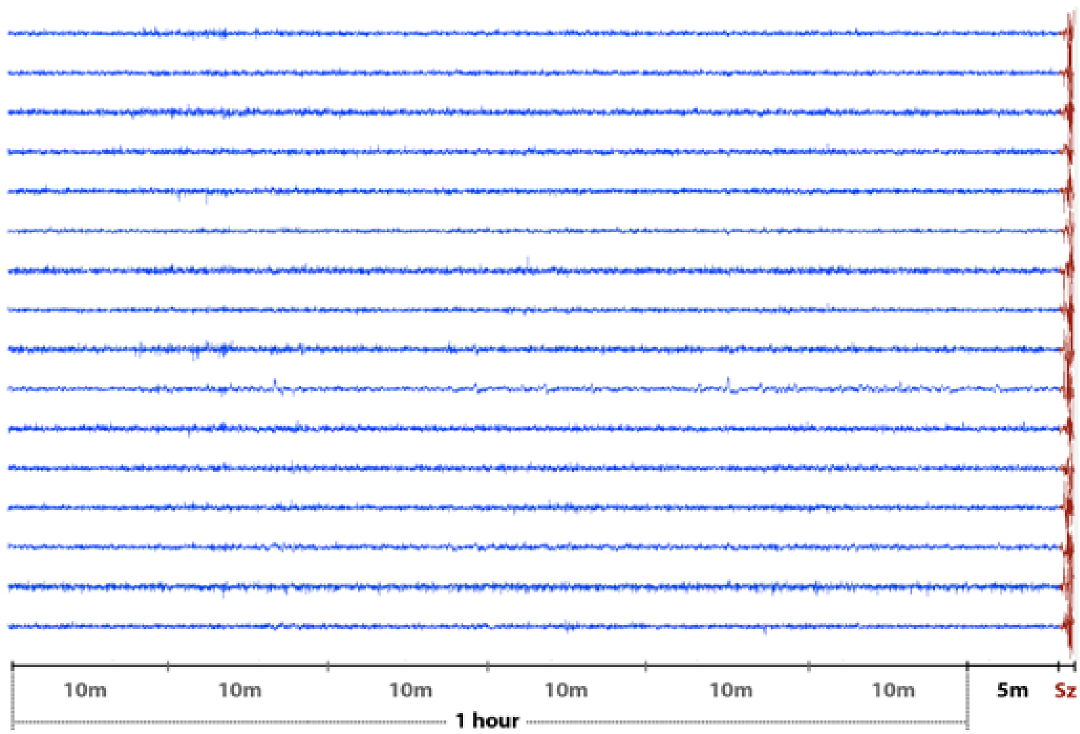


Figure 2.1 One hour of preictal segment

In the same way, one hour sequences of interictal ten minute data sections are provided here. However the interictal data were picked from full data records arbitrarily, with the constraint that interictal segments be as far from any seizure as can be practically possible for avoiding 13 confusion between postictal and preictal signals. In the human records interictal segments were taken more than four hours after or before seizure onset.

b) Data Preprocessing

The dataset is composed of.mat files each containing 10 minutes of 15 channel readings. For simplicity, we train on one patient's data and assume that each channel is equally well representative of the signatures we expect to learn in classifying whether the segment precedes a seizure event or not. We ultimately want to run this model on a resource limited wearable or build a server which can make fast responses to the requests, we sample 1 second segments from a single channel before computing the spectrogram using scipy.signal. This lets us use powerful and effective CNNs on the vision problem after applying fourier smoothing over the noisy and nonstationary signal over 1 second time windows, to fit on devices as small as an Arduino. Spectrogram: A spectrogram is a visual method for depicting the strength of the signal, or "loudness", over time at various frequencies present in that particular waveform. A spectrogram can be generated by an optical spectrometer, by band pass filters, by fourier transform or by a wavelet transform. In this project we will be using fourier transform to create the spectrograms out of raw eeg data. We use the scipy library to create spectrograms which are built using a short time fourier transform and a log is taken over it to increase its intensity. This gives us a visual way to look at our signal data which can now be classified using deep learning vision models such as convolutional neural networks. On visualizing the resulting spectrogram we observe that there is a lot of difference in the looks of the interictal and preictal segments which gives us the confirmation that the problem is successfully converted into a vision problem from a generic signal processing problem.

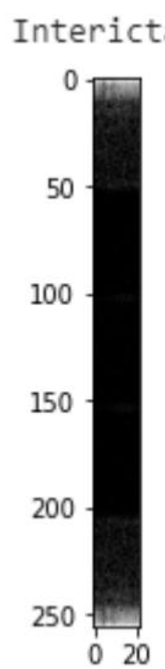


Fig. 2.2 One second worth of data from channel 0 of Patient1's Interictal segment size (256,22)

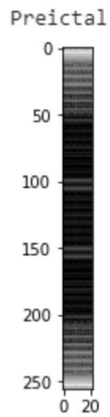


Fig. 2.3 One second worth of data from channel 0 of Patient1's Preictal segment size (256,22)

We reshape this into (128,128) size for obtaining some symmetry for improving the performance of convolutions after reshaping our images look like following:

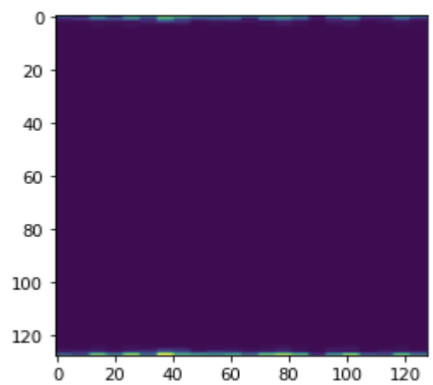


Fig. 2.4 An Interictal channel 0 sample reshaped to (128,128)

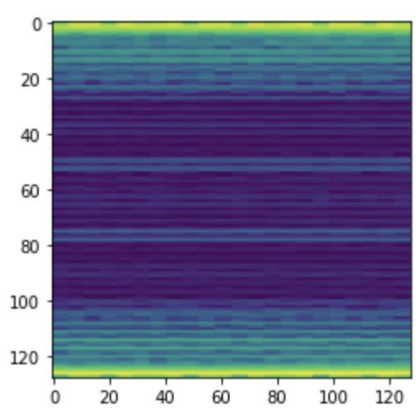


Fig. 2.5 An Preictal channel 0 sample reshaped to (128,128)

We make use of opencv's interpolation while reshaping. Resizing an image needs a method to calculate the value of the pixel for the new image from the original image. We make use of INTER_AREA interpolation. The final step to our preprocessing is stacking these images in 3d axis i.e the z axis, though any number of channels can be stacked we kept it limited to 3 for simplicity, here we assume that 16 each channel is equally representative and can help us classify between interictal and preictal segments.

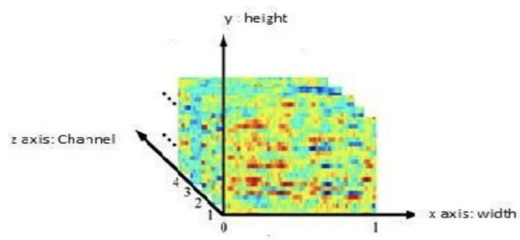


Fig. 2.6 Input to cnn containing multiple channels (128,128,n)

c) Training

Now we take our preprocessed images and feed them to our convnets so that it can learn to classify between two classes preictal and interictal, our input shape to the model is (128,128,3) while the shape of the output of the neural network is (2,) representing the probability of each class. In our training set the interictal class is represented as 1 while the preictal class is represented as 0. We trained models such as efficient net, resnet, inception and inception-resnet which use imagenet pretrained weights and hence we employ a sort of transfer learning in our approach however real transfer learning kicks in when we use weights from one patient's model to train for another patient.

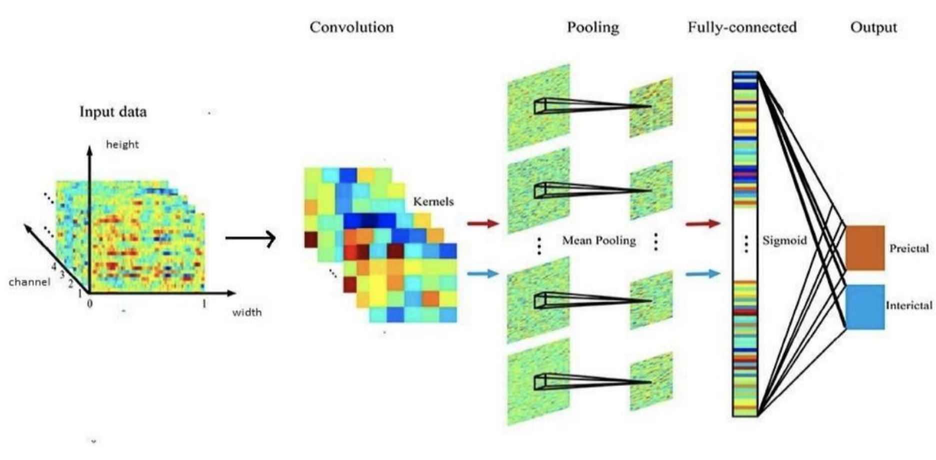


fig training procedure

Fig. 2.7 Training Procedure

Components of our training procedure are as follows:

1. **Convolution:** A CNN is capable of capturing the Space-based dependencies in an image through the use of relevant filters. This model performs a better convergence to the image dataset as the parameters can be shared and are sparse, hence training takes lesser computation power compared to simple densely connected neural network techniques. In other words, the network can be trained to understand the intricacies of the image better. The convolution sequence is a feature detector as it extracts the meaning out of the images by attaching some layers on top of it so we can build powerful classifiers.

2. **Pooling:** It is quite similar to Convolutional Layer in many ways like having a filter and running it through the image's height and width dimension. The Pooling layer is liable for reducing the space shape of the Convolved output. This is often to decrease the computational power required to process the information through dimensionality 18 reduction. Furthermore, it's useful for extracting important features which are rotation and position invariant, thus maintaining the method of properly training the model. There have been arguments on why pooling should not work as it throws away a lot of information but experiments have proved that even though it's counter-intuitive pooling works and improves the performance of CNNs. There are two ways of Pooling: Max Pooling and Average Pooling. Max Pooling calculates the most value from the section of the image covered by the Kernel. Whereas, Average Pooling calculates the average of all the values from the sections of the image covered by the Kernel. Max Pooling also doubles as a Noise Suppressant.

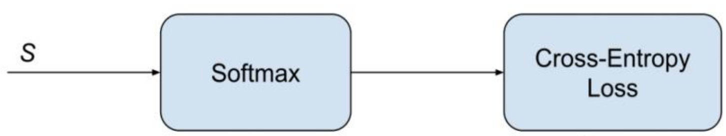
It removes the noisy activations altogether and also performs noise removal together with dimensionality reduction. On the opposite hand, Average Pooling simply performs dimensionality reduction as a noise reducing mechanism. Hence, we are able to say that Max Pooling performs a lot better than the generic Average Pooling. The Convolutional Operation and the Pooling operation, together constitute one layer of a Convolutional Neural Network. looking at the intricacies within the images, the quantity of such layers could also be increased for capturing low-levels details even further, but at the expense of more and more computational power. After researching the above process, we've successfully enabled the model to know the features. Moving on, we take the flattened output and feed it to a daily Neural Network for classification purposes. After going through the above process, we have successfully enabled the model to extract and understand the features. Moving on, we will flatten the final resulting array and feed it to a regular classifier for classification purposes.

3. Fully-Connected: A Fully-Connected layer is usually applied to learn non-linear combinations of the high-level features as represented by the output of the convolutional layer. The Fully-Connected layer is learning a possibly non-linear function in this scenario. 19 4. Softmax: Now we've converted our input image into an appropriate form and features have been extracted using convolutional layers, we should now flatten the image into a column vector. The flattened output is given into a feed-forward neural network and backpropagation applied to each iteration of coaching. Over a series of epochs, the model is in a position to differentiate between dominating and certain low-level features in images and classify them using the Softmax Classification technique.

$$\mathbf{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$

4. Loss: we make the use of Categorical cross entropy as our loss function. it is a loss that is employed for single label categorization. This is when only one category is applicable for each data example. which means that an example can be of one class only.

5. Learning rate schedulers: The learning rate is a configurable hyperparameter used in the training of neural networks that has a small positive value, it is in the range between 0.0 and 1.0. The learning rate adjusts how quickly the model learns to solve the problem. 20 Learning rate is arguably one of the most important hyperparameters in a neural network.



$$f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}} \qquad CE = -\sum_i^C t_i log(f(s)_i)$$

A learning rate scheduler is the scheme following which the learning rate is adjusted during the training procedure. We employed two types of learning rate schedulers they are:

1. Reduce Learning rate on Plateau: This learning rate scheduler basically monitors or checks a particular metric ex validation loss for improvement if there no improvement is seen for a set of number of steps, learning rate is decreased by a pre specified factor and the process continues, there are options to set minimum learning rate beyond which the learning rate will not be reduced. Models perform better if learning rate is reduced by the factor of 2-10 once learning plateaus. This procedure checks a quantity and if no improvement is seen learning rate is reduced after a 'patience' number of epochs.

2. Cosine annealing: The cosine annealing learning rate scheduler is an example of a very aggressive learning rate scheduler where learning rate starts high and is dropped relatively 21 rapidly to a minimum value near zero before being increased again to the maximum. This helps in escaping local minima so that convergence to the global minima could be made possible.

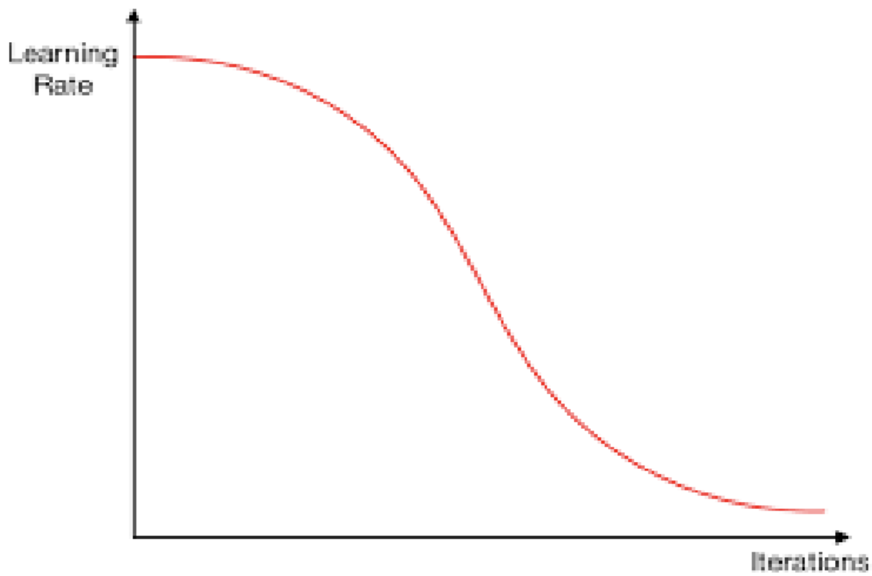


Fig. 2.8 Learning curve rate

We applied the following pretrained models to learn the features, these can be used separately or their output can be assembled for better accuracy.

1. Resnets

They employ the use of skip connections to solve vanishing and exploding gradients problems previously it was difficult to train deep neural networks that are very deep (containing many layers) as the activation would become very small (vanishing gradients problem) or become too large (exploding gradients problem).There are many variants of resnet based on number of layers namely resnet34, resnet50, Resnet101 and Resnet 152. For simplicity and due to limited computational power we used resnet50 for our experiments the Resnet Model block architecture looked like the following.

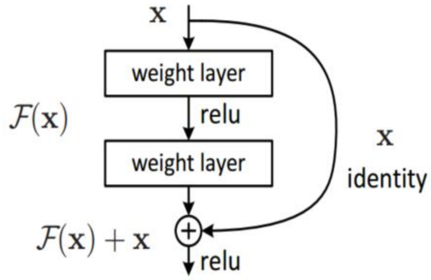


Fig. 2.9 Architecture of Relu Function

Our architecture looks like following:

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
resnet50 (Model)             (None, 2048)              23587712
_________________________________________________________________
dense_1 (Dense)              (None, 758)               1553142
_________________________________________________________________
dense_2 (Dense)              (None, 512)               388608
_________________________________________________________________
dense_3 (Dense)              (None, 256)               131328
_________________________________________________________________
dense_4 (Dense)              (None, 2)                 514
=================================================================
Total params: 25,661,304
Trainable params: 25,608,184
Non-trainable params: 53,120
```

Results Resnet:
Accuracy_score : 0.67166

Confusion matrix:

n=1200	**Predicted:** YES	**Predicted:** NO
Actual: YES	289	311
Actual: NO	83	517

Precision: [class 0 : 0.77688, class 1: 0.62439]
Recall: [0.48166, 0.86166]
Fbeta_score: [0.59465, 0.72408]

So here we can see that resnet50 does not perform really well the false positive rate is pretty high as well, the precision for class 1 i.e interical is very low as well(higher values are preferred).

2. Inception Resnet v3

The Inception network was a major achievement in the development of CNN. Before it most CNNs just appended convolution layers one after another to get better performance.

Inception network applies multiple kernels with differing size ex (3,3) (5,5) (7,7) and so on, the padding is kept same so that the output is of the same size for all the kernels of different size then these outputs are concatenated and joined to make the output of the convolution operation. Inception resnet network however uses the utility of the skip connections in the inception network concept making it more robust and scalable. Inception network architecture looks like:

Layer (type)	Output Shape	Param #
inception_resnet_v2 (Model)	(None, 1536)	54336736
dense_5 (Dense)	(None, 758)	1165046
dense_6 (Dense)	(None, 512)	388608
dense_7 (Dense)	(None, 256)	131328
dense_8 (Dense)	(None, 2)	514

Total params: 56,022,232
Trainable params: 55,961,688
Non-trainable params: 60,544

Results Inception_Resnet:
Accuracy_score: 0.85416
Confusion matrix:

n=1200	Predicted: YES	Predicted: NO
Actual: YES	523	77
Actual: NO	98	502

Precision: [0.84219, 0.86701]
Recall: [0.87166, 0.83666]
Fbeta_score: [0.85667, 0.85156]

So here we can see that inception_resnet _ v3 drastically improves the performance we see higher precision and recall as well as accuracy. This is because of the complexity of the inception_resnet model. False positive rate however didn't improve much.

Efficient Net b3

EfficientNet can be considered a group of convolutional neural network models. But given some of its subtleties, it's actually more efficient than most of its predecessors.

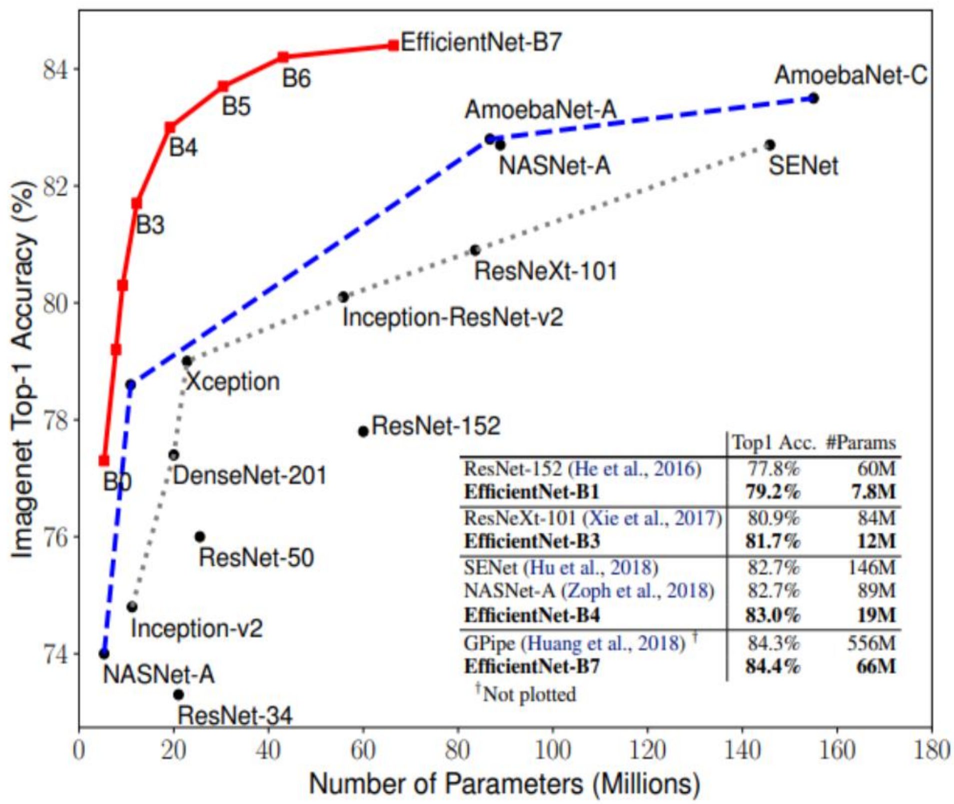


Fig. 2.10 Parameter vs Accuracy graph

The EfficientNet model group consists of 8 models from B0 to B7, with each subsequent model number referring to variants with more parameters and higher accuracy. In the efficient net architecture scaling is done on depth, width, and resolution—focusing on all three in combination has made for better results. We decided to use efficient net b3 due to our limited computational resources however Accuracy Is bound to increase if we use more complex efficient nets such as b5 and b7 however they may also need proper regularization to provide best fit.

Our b3 architecture looks like:

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
imgs (InputLayer)            [(None, 128, 128, 3)]     0
_________________________________________________________________
tf_op_layer_Sub_1 (TensorFlo [(None, 128, 128, 3)]     0
_________________________________________________________________
tf_op_layer_RealDiv_1 (Tenso [(None, 128, 128, 3)]     0
_________________________________________________________________
efficientnet-b3 (Model)      (None, 4, 4, 1536)        10783528
_________________________________________________________________
global_average_pooling2d (Gl (None, 1536)              0
_________________________________________________________________
dropout (Dropout)            (None, 1536)              0
_________________________________________________________________
dense (Dense)                (None, 2)                 3074
=================================================================
Total params: 10,786,602
Trainable params: 10,699,306
Non-trainable params: 87,296

_________________________________________________________________
```

Results Efficient B3

Accuracy_score: 0.89333

Confusion matrix

n=1200	Predicted: YES	Predicted: NO
Actual: YES	542	58
Actual: NO	70	530

Precision: [0.88562, 0.90136]
Recall: [0.90333, 0.88333]
Fbeta_score: [0.89438, 0.89225]

So here we can see that b3 improves the performance we see higher precision and recall as well as accuracy. The false positive rate is also lowest which makes b3 a good choice for a use cases like this. Efficient Net b3 is way more compact then inception_resnet and hence it can work well if used for serving or in an embedded device.

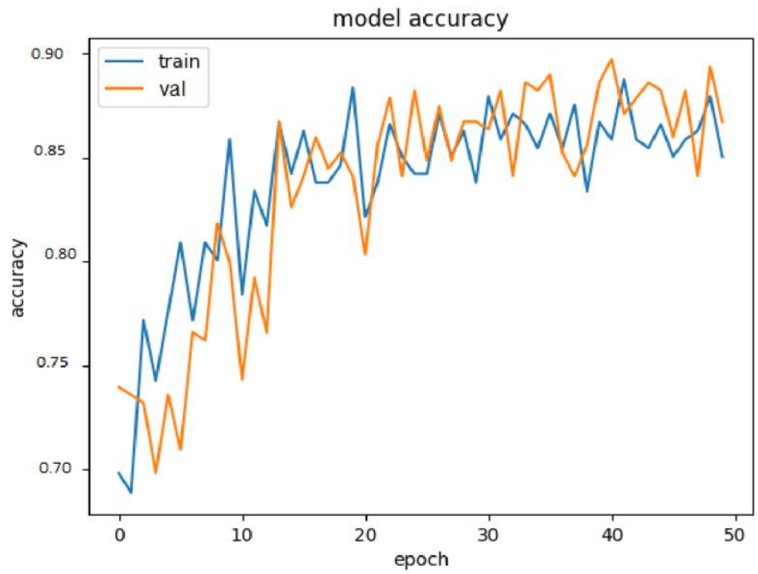


Fig. 2.11 Model Accuracy Graph

Accuracy graph plotted against increasing number of epochs, train and val accuracy plotted.

Graph plotted for efficient net b3 model.

This graph is the most used plots to plot Loss curve during training to debug a neural network.

It gives us a quick visualization of the training process and the direction of the network.

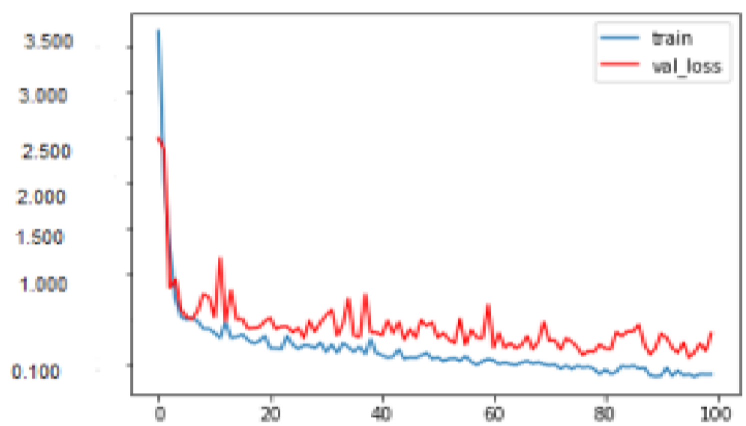


Fig. 2.12 Model Loss Graph

Accuracy graph plotted against increasing number of epochs, train loss and val loss plotted. Graph plotted for efficient net b3 model.

Conclusion and Future

In this study we used CNNs to detect seizure and generate alerts before a seizure event.

We successfully framed the problem into a Computer vision problem and used CNNs to classify Interictal and preictal segments with high accuracy and low false positive rate. Models were built considering computation constraints of small devices where these models can be possibly Deployed.

When the data set is small there are chances of very high variance which means we get different sets of weights every time we train our neural network which produces different results or predictions. In these conditions Ensembling is used to reduce the variance of the neural network.

It is done by training multiple models instead of single model and then combine the prediction form all these models. These reduces the variance (bragging), bias (boosting) and improve prediction (stacking) in a neural network.

Bagging: It means bootstrap aggregation. Reducing variance by averaging out weights.

For example we can train N trees on different subset of data and perform ensembling on It.

Boosting: These algorithms convert weak learners to strong learners. Boosting involves fitting of a sequence of weak learners models that are better than random guessing like small decision trees to weighted versions of the data. More weights are updated to misclassified by earlier rounds. Unipolar and bipolar modes are used to record EEG signals. In polar mode the voltage difference between all electrodes and a reference is recorded, where due to electrode-reference pairs a channel is formed. On the other hand in bipolar mode, voltage differences between two electrodes are recorded and here each pair becomes a channel. There are many evaluation techniques which are used like filtering techniques, wrapper techniques, embedded techniques and hybrid techniques.

EEG datasets have lot of noise in it and usually it is due to activity such as blinking and movement of eyes, breathing etc. Some but not all such activities can be controlled and `Artefacts can be reduced and EEG recordings can be taken. Lesser artefacts present in

EEG recording can improve prediction accuracy drastically. EEG data is important for neuroscience and used extensively for brain-machine interfacing. Invasive EEGs are the recording which takes place by placing electrodes directly into the brain. It is the most accurate and much less noise/artefact. Artefact is caused mainly due to blinks and eye movement, breathing etc. Targeted readings can be taken from invasive EEG. Whereas non-invasive EEG can be performed very easily at home using a helmet like device. It is prone to noise. Data obtained from Invasive EEG will provide higher accuracy.

References

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2016). Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., & Rabinovich, A. (2015). Going deeper with convolutions. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 1-9.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770-778.

Zhou, M., Tian, C., Cao, R., Wang, B., Niu, Y., Hu, T., & Xiang, J. (2018). Epileptic seizure detection based on EEG signals and CNN. *Frontiers in neuroinformatics*, *12*, 95.

Wei, X., Zhou, L., Chen, Z., Zhang, L., & Zhou, Y. (2018). Automatic seizure detection using three-dimensional CNN based on multi-channel EEG. *BMC medical informatics and decision making*, *18*(5), 111.

Roy, Y., Banville, H., Albuquerque, I., Gramfort, A., Falk, T.H., & Faubert, J. (2019). Deep learning-based electroencephalography analysis: a systematic review. *Journal of neural engineering*, *16*(5), 051001.

Acharya, U.R., Oh, S.L., Hagiwara, Y., Tan, J.H., & Adeli, H. (2018). Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals. *Computers in biology and medicine*, *100*, 270-278.

Ahammad, N., Fathima, T., & Joseph, P. (2014). Detection of epileptic seizure event and onset using EEG. *BioMed research international*. https://doi.org/10.1155/2014/450573

Alotaiby, T., Abd El-Samie, F.E., Alshebeili, S.A., & Ahmad, I. (2015). A review of channel selection algorithms for EEG signal processing. *EURASIP Journal on Advances in Signal Processing* (1), 66.

Craik, A., He, Y., & Contreras-Vidal, J.L. (2019). Deep learning for electroencephalogram (EEG) classification tasks: a review. *Journal of neural engineering*, *16*(3), 031001.

Eberlein, M., Hildebrand, R., Tetzlaff, R., Hoffmann, N., Kuhlmann, L., Brinkmann, B., & Müller J. (2018). Convolutional neural networks for epileptic seizure prediction. *In IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2577-2582.

Rasheed, K., Qayyum, A., Qadir, J., Sivathamboo, S., Kwan, P., Kuhlmann, L., & Razi, A. (2020). Machine Learning for Predicting Epileptic Seizures Using EEG Signals: A Review. *arXiv preprint arXiv:2002.01925*.

Truong, N.D., Nguyen, A.D., Kuhlmann, L., Bonyadi, M.R., Yang, J., & Kavehei, O. (2017). A generalised seizure prediction with convolutional neural networks for intracranial and scalp electroencephalogram data analysis. *arXiv preprint arXiv:1707.01976*.