A Deep Learning Model for Irish English and Hindi Language Identification

MSc Research Project
Data Analytics

Sumit Singh
Student ID: x20135769

School of Computing
National College of Ireland

Supervisor I:	Dr. Paul Stynes
Supervisor II:	Dr. Pramod Pathak

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Sumit Singh
Student ID:	x20135769
Programme:	Data Analytics
Year:	2022
Module:	MSc Research Project
Supervisor:	Dr. Paul Stynes
Submission Due Date:	31/01/2022
Project Title:	A Deep Learning Model for Irish English and Hindi Language Identification
Word Count:	6083
Page Count:	17

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	31st January 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A Deep Learning Model for Irish English and Hindi Language Identification

Sumit Singh
x20135769

Abstract

Spoken Language Identification is a process of recognizing languages based on audio samples. Language identification models help enable speech-based applications that ease the use of technology for people who find modern technology challenging. This research proposes a deep learning language identification model that can identify and differentiate between three languages Irish, English and Hindi. The proposed deep learning model performs feature extraction based on the frequency and pitch of an audio sample represented by a mel spectrogram using a convolutional neural network (CNN). The audio samples are of varying sizes from 1 second to 10 seconds. The audio sample are used to create the RGB spectrum-based spectrograms. The spectrograms are processed using data augmentation techniques. Pre-trained models such as Resnet50, InceptionV3, EfficientNet-B0 are also trained along with the proposed CNN model and evaluated based on the accuracy, recall, precision and loss values. The CNN model achieves best with an accuracy of 93.50% on the test dataset. This research will help create faster and more efficient speech-based applications in Irish, English and Hindi languages.

1 Introduction

Spoken language identification has been a keen interest for the AI community in the recent past. The key reason is the numerous applications of language and speech recognition technology. With the evolution of digital assistants and speech-enabled devices, the world is becoming easy to navigate for the majority of people. Instead of performing tasks manually, speech and voice controls can be leveraged to navigate through the majority of tasks. From voice search on the phone to operating cars using voice control, the possibilities are endless.

The aim of the research is to differentiate amongst languages with the same linguistic background from the Indo-European Language Family. The major contribution of this research is a novel deep learning based CNN model to differentiate between the Irish, English and Hindi languages using mel spectrograms for feature extraction. The creation of a language identification system is highly dependent on efficient audio and image processing. Image and audio processing has mostly been done using Deep Neural Networks (DNN) and Recurrent Neural Network (RNN). Although the results are promising, there is a gap in extracting features that can enable more accurate differentiation between the languages due to the statelessness or vanishing gradient problems of the neural networks as explored by Basodi et al. (2020). Many researchers have tried rectifying problems using Long Short-Term Memory (LSTM) models that can remember the state of the weights of the model training and support backpropagation techniques. However, more accurate determinations can be made using convolutional neural networks (CNN) as the network supports a variation of multilayers perceptron's and can deliver precisely fixed size outputs. CNN has delivered promising results in the field of image processing, recognition and classification and provides an interesting proposition to solve challenges faced in speech and audio processing by identifying relevant features of pitch, amplitude and frequency using the spectrograms images of the audio samples.

A spectrogram is a visual representation of the spectrum of frequencies of an audio signal. A spectrogram often known as the vanilla spectrogram is a two-dimensional graphical representation of the frequency of an audio signal at a point in time. The strength of the speech at a point in time is denoted by the colour of the signal. These colour differences can help identify the different patterns in a voice sample and can help train neural networks to identify different features that can be used to differentiate between sounds. The frequency for a spectrogram is defined in hertz and is linear by nature. The linear frequencies can help differentiate between sounds, but humans perceive audio frequencies at a much lower frequency scale as compared to other lifeforms on earth. Therefore, the language identification systems need to be designed based on the audio frequency register spoken and understood by human ears. Therefore, the need to create spectrograms in a logarithmic scale is essential to process human languages that can help identify better features to differentiate between languages.

Humans perceive frequencies logarithmically and to create a system that accurately identifies different languages, it is essential to create spectrograms based on a logarithmic scale so that the created system is more perceptually relevant to human languages. Mel scale is such a logarithmic scale where the perceptual distance between two frequencies at two different frequency registers is the same and is defined by a basic unit of measurement known as mels. The pitch of two different audio samples, when perceived in hertz by linear frequency, will be different, however, the same voice samples when perceived in mels rather than hertz will have the same pitch at higher and lower ranges of frequencies. Therefore, the frequency bands are equally spaced on a mel scale and the response is more relevant for human auditory systems as compared to the hertz linearly spaced frequency bands. Hence, for the research mel spectrograms are created for the audio signals and processed using CNN to extract features for the language identification system. The relation between mels and hertz is logarithmic and is defined by

$$m = 2595 \cdot log(1 + \frac{f}{500})$$

Wherein, the f is the frequency in hertz while 2595 is the value of constant C calculated by log with base 10 in order to correspond 1000 hertz equivalent to 1000 mels. The conversion of hertz frequency to mels is also dependent on different hyperparameters that are defined by mel filter banks. The parameters help decide the level of feature extraction and are dependent on the quality, pitch and noise levels of the audio samples. Different parameters were tested and a final set of parameters were chosen to create the mel spectrograms for the experiments.

The paper discusses creating convolutional neural networks to examine and evaluate mel spectrograms created using audio samples from different speakers speaking Irish, English and Hindi Languages. The existing techniques will be explored in related work

Section 2. The research methodology used has been discussed in Section 3, followed by the design specification of the proposed model in Section 4. The implementation of the model is discussed in Section 5. Evaluations have been captured and discussed in Section 6 while Section 7 concludes the research followed by the future work.

2 Related Work

Language identification model creation requires a detailed survey of historic language and speaker identification models in Section 2.1, existing spectrogram processing and feature extraction techniques in Section 2.2 and CNN architectures in Section 2.3.

2.1 Language Identification Models

Over the time period, different approaches have been taken to develop a language identification system (LID). Most of the approaches have been involved around the use of the i-vectors system. The system was initially designed for speaker recognition using Gaussian mixture models (GMM) super vectors by calculating Maximum a Posteriori(MAP) probability to explain the distribution of weights in a dataset. The approach was later redesigned by Lopez-Moreno et al. (2014) to be implemented in a language recognition system. In the proposed model features were extracted with a frame rate of 10ms over 25ms long training windows. The universal background model(UBM) was trained with 1024 components and refined over 10 iterations. I-vector was a feature extraction method where weights of each language were captured and stored, and the classification was performed using different classification architectures. The researchers developed a linear feed-forward deep neural network with the output being configured by a softmax function and a cross-entropy cost function. The network was trained on two separate datasets and the evaluation concluded that DNN when coupled with i-vector for feature extraction, resulted in better results when compared to using i-vector with logistic regression.

The approach was further refined by Heracleous et al. (2018) by combining the i-vector system with Convolutional Neural Network and Deep Neural Network. The researchers created two separate models for both networks and evaluated the model based on the equal error rate(EER) of the language classification. Amongst the 50 evaluated languages, the combination of CNN with the i-vectors model delivered the best performance of 3.48% EER as compared to 3.55% EER of the DNN model. Although i-vector seems to provide promising results, vectors are hard to manage for large datasets because of the large weights of the features.

Therefore, a system based on word embeddings was designed by Lozano-Diez et al. (2018), Embeddings are fixed length i-vector systems that capture the whole utterances and can store more relevant weight information for language identification. The researchers created a Bidirectional-LSTM model in combination with DNN and RNN (DNN-BLSTM-RNN) model. The model is followed by a pooling layer which is in turn connected to two fully connected layers, the layers correspond to the embeddings and a softmax output layer is followed by a multi-class cross-entropy function to distinguish between the languages based on posterior probabilities. The classification of 20 languages indicated that the performance of the model is comparable to the i-vector system and if trained over a larger set of languages the model can even outperform state of the art system.

Bartz et al. (2017) takes a newer approach by training Recurrent neural networks(RNN) on input features like Mel Frequency Cepstral Coefficients (MFCC) instead of using i-vector systems. The researchers conclude that the system can be trained as efficiently as the state-of-the-art system but with lesser complexity. The researchers create a CNN model to extract the features of the input image and the output layer is fed to the RNN-Bidirectional-LSTM for classification. The network uses ReLu activation followed by batch normalization. The system also employs transfer learning using InceptionV3, the results are promising for the base model but are elevated after using the transfer learning method. China et al. (2018) explores the creation of mel-spectrograms to feed into the CNN model for feature extraction. A combination of the CNN-LSTM-RNN model is used to analyze two sets of spectrograms, a pitch-chroma spectrogram and mel-spectrograms. The system is tested over nine different languages with two different sets of samples size. The samples are created using audio recordings of 3 seconds and 10 seconds. Similar to the state-of-the-art systems, the model uses the ReLu activation function however, it uses Adadelta optimizer instead of standard Adam optimizer. Mel spectrograms show higher accuracy over chroma spectrograms when tested over the NITS-LD dataset.

Arla et al. (2020) further analysed the system by using mel spectrogram to create a multi-class language identification system for four different Indian languages. The feature identification is done using MFCC coefficients and trained using the CNN network. The researchers create a model sequential CNN model without employing any transfer learning methods and achieve an accuracy of **88%**. The model employs ReLu activation combined with an SGD optimizer.

2.2 Spectrogram, Pre-processing and Feature Extraction Techniques

Based on the research of techniques used by state-of-the-art systems, the process of using raw audio signals as a base for creating mel spectrograms and extracting features seems to be the optimal way to create an efficient language identification system. However, it is crucial to analyze different techniques for spectrogram creation and feature extraction.

Purwins et al. (2019) discusses various techniques for audio signal processing. Researchers create mel-spectrograms and use different feature extraction techniques including MFCC and DNN models to identify relevant features for automatic speech recognition. The approach using neural networks provide better results as compared to MFCC along with DCT. Log-spectrograms are created using filter banks and different window sizes to serve as an input to CNN models. Dörfler et al. (2017) further concludes that spectrograms are a better source of feature extraction as raw audio data can be more structurally mapped for feature extraction. Different size audio samples of two and four seconds are considered with a sampling rate of 22050Hz and a window size of 2048 with a shift parameter of 512 samples. Researchers conclude that higher frequency regions rarely contain more information. Therefore, the mel scale is the optimum choice for creating spectrograms. However, after convolution and pooling, the long and short window spectrograms performed equally well. Hall et al. (2019) performed multi-class instrumental audio classification of different musical instruments wherein after removing three classes of instruments, the accuracy of the model was increased. Therefore, it is crucial to create a classification report for the final results in order to analyze whether the three languages in the experiment have a linguistic similarity.

Feature extraction is also dependent on the pre-processing of the spectrogram. Therefore, a key insight into the properties of the spectrogram is essential. Stolar et al. (2018) experiments to assess the effectiveness of the RGB colour spectrum of the spectrogram.

RGB colour spectrograms are generated and analysis of the effect of each of the three colours is performed. Spectrograms are distributed into four different frequency scales linear, mel, log and equivalent rectangular bandwidth(ERB). Based on evaluation parameters such as accuracy, recall, precision, and F-score Mel-RGB spectrogram outperformed any unicolour spectrogram indicating RGB spectrograms are a better choice for feature extraction. Meghanani et al. (2021) defines different hyperparameters for processing and creation of spectrograms by generating mel spectrograms using 224 filter banks with a Hanning window of 2048 samples and a hop length of 512 samples. The effective performance was achieved using an image size of 224 X 224 X3 which is also the dimension used by various transfer learning models like Resnet. The Resnet model with mel-spectrograms achieved an accuracy of 62.5% on the dataset. The RMSE for the model was improved by 2.6% from the base model.

After pre-processing, feature extraction is the most crucial part for accurate language identification. Amongst different techniques, Shintri and Bhatia (2015) explores the feature extraction process using mel-frequency cepstral coefficient (MFCC) computed using a windowed discrete Fourier transform (DFT). The process requires the creation of frames of the speech samples using windows of different lengths. Post division of signals, an average of the spectrum is computed and helps decide values for the mel banks that improve the equal error rate and helps in stable hyperparameter settings. Logarithmic nonlinearity is then removed using normalization methods such as mean and variance normalization (MVN) and frequency wrapping.

The process is mostly manual and requires complex calculations. Therefore, it is crucial to explore the comparison between feature extraction methods using MFCC and DFT with the creation of Mel spectrograms. Demircan and Örnek (2020) performed a comparative study for emotion classification using both feature extraction techniques. Two applications were designed wherein the first application spectrogram images were classified using a CNN architecture and a second application using manually extracted MFCC features implemented by a DNN model. The researcher concluded that the features extracted through spectrogram images were more discriminative as compared to manual features extracted and processed using DNN.

2.3 CNN Architectures

Convolutional neural networks (CNN) are a form of neural network models that can be created using different architectures. A survey of different CNN architectures is essential to highlight the important aspects that help in designing an efficient language identification framework.

Kaiyr et al. (2021) explores a combination of CNN and LSTM architectures with 117,703 trainable and 416 non-trainable parameters to create a language identification system for seven different languages. The architecture uses four convolutional layers followed by ReLu activation with different kernel sizes and 3 X 3 filters. Each layer is followed by batch normalization and dropout to reduce overfitting. The output layer is implemented using SoftMax activation and cross-categorical entropy for loss function. Bohra and Bhatnagar (2021) creates a similar architecture including pooling layers, wherein the first two layers are responsible for extracting the features from the input signals while the third layers reduce and identify the number of relevant features, the last layer is a fully connected layer which is used for backpropagation addressing the issue of vanishing gradients. The output layer is optimized using a SoftMax function followed by different optimizers. Ideally, RMSprop serves as an optimum option as the learning rate for the model can be pre-defined and optimized for maximum efficiency. Apart from custom made CNN architectures, it is imperative to explore different pre-trained architectures. ResNet network (ResNet) is one such architecture that addresses the problem of vanishing gradients using skip connections and connecting two non-adjacent layers.

Observations made by Revay and Teschke (2019) indicate that using pre-trained Resnet50 architecture the performance of the language identification system can be marginally increased. Celano (2021) designed experiments to compare the performance of a 3-layer sequential CNN model and a ResNet 50 pre-defined model. The performance of the models is similar, however, Resnet outperforms the custom CNN model as it can capture loss properly.

With the ResNet network performing well, it becomes an interesting proposition to take a look at and analyze different pre-trained models. Singh et al. (2021) performs a detailed analysis of different architectures for spoken language identification. Various architectures like a traditional CNN model, Resnet50 and InceptionV3 models were compared. Amongst all the architectures, traditional CNN and InveptionV3 provided the best results. The CNN model was created using a 2D convolutional layer followed by normalization and pooling layers. The model was trained to 60 epochs with a batch size of 32 with a relu activation function. Adam optimizer was used with the SoftMax function in the output layer.

Similarly, Lu et al. (2020) performs speech recognition tasks using the EfficientNet model where mel spectrograms were created for feature extraction. EfficientNet is a scaled-up version of the ConvNets that uses a fixed set of scaling coefficients. The architecture has different versions with EfficientNet-B0 having 237 layers and EfficientNet-B0 having 813 layers. For the experiment, with limited computation power, it makes more sense to use EfficientNet-B0 for training and testing the dataset. Since the experiment is performed using a single GPU, it is essential to examine the computational benchmark of different pre-trained versions. Bianco et al. (2018) performs an in-depth analysis for the majority of DNN networks and performs an operational and computational benchmarking of different pre-trained networks for image processing. Some of the key findings of the paper are that the complexity of the network is not a key benchmark for the efficiency of the network and the recognition efficiency of the models is not dependent on the number of operations. Based on the analysis of the top-one vs top-five accuracy levels depending on GPU power; VGG, Resnet and Inception architectures are the optimum choices for the experiment.

To summarize, based on the analysis. CNN was used instead of RNN and other DNN deep learning methods for model creation. Mel spectrograms were created for automatic feature extraction instead of manual feature extraction using DNN. Different techniques like Pooling, Regularization, Image augmentation and dropout will be used to create the novel method. A detailed analysis of hyperparameters for mel spectrogram helped identify parameters required to create more informative mel spectrograms. The state of the art experiments consider a multi class approach by assuming that all the languages are mutually exclusive. However, all languages have common roots and belong to certain families and similar accents of the speakers also provide high chances of language misclassification. Therefore, the experiments are designed for a multi-label classification wherein the probability of each predicted language is calculated and one with the maximum probability is chosen.

3 Methodology

The research methodology consists of 5 steps, data gathering and conversion, spectrogram creation, spectrogram pre-processing, data modelling, evaluation and result.

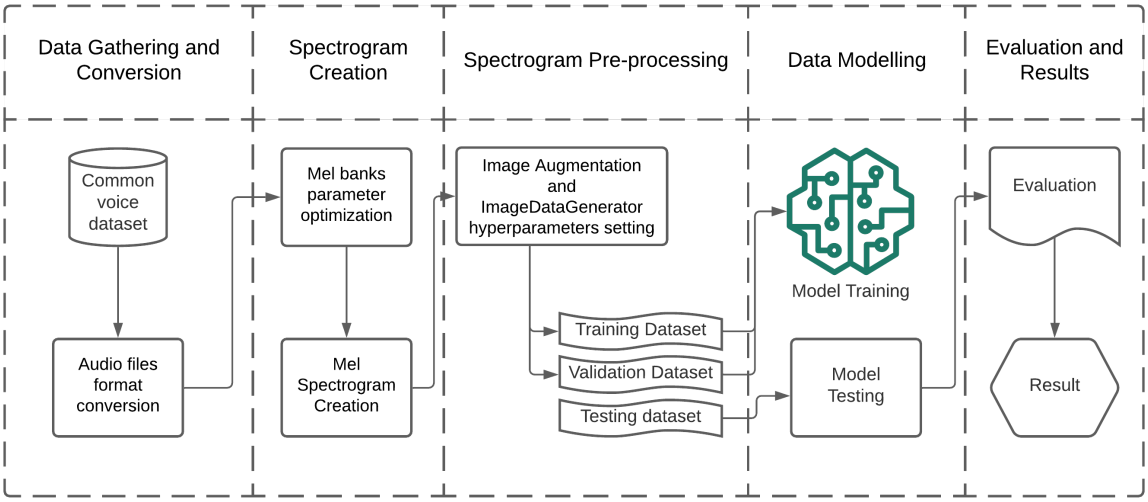

Figure 1: Research Methodology

The first step, data gathering and conversion involve the collection of audio files from the Mozilla commonvoice dataset [1]. The dataset contains different common voice corpora for Irish, English and Hindi amongst other languages. Each dataset consisted of over 10,000 voice samples. Based on audio quality and noise levels, 1000 samples were segregated in a separate folder. The datasets contained voice samples from 1 second to 10 second time duration and are in an mp3 format. Using the AudioSegment() function of the pydub python library, the audio samples were converted to .wav which is a pre-requisite for the creation of mel spectrograms using the Librosa library.

The second step, spectrogram creation involved the creation of mel spectrograms using the Librosa python library. To create the spectrogram, the first step was to define the hyperparameters for mel banks. Based on the audio samples in the dataset mel banks were created with a sample rate of 22050, a frequency bin(n_fft) of 2048, hoping length(hop_length) of 512 and number of bands(nmels) for 128. The shape of the spectrogram is defined with (nmels, number of extracted features). The spectrograms for the three languages are combined and stored in a folder to be processed using the CNN model.

The third step, spectrogram pre-processing was performed using a Keras pre-processing function known as ImageDataGenerator. The function also provides pre-processing abilities like rescaling, resizing, flipping. The colour mode set for the spectrogram is RGB. Three different generators are created training, validation and testing datasets. The training set contains 2000 images, the validation set contain 600 images and the testing set contains 400 images. The images were rescaled to 1./255 and a batch size of 20 is defined for training and validation datasets, while the testing dataset is processed with a batch size of 1 to ensure every image is tested using the CNN model.

[1] Dataset: `https://commonvoice.mozilla.org/en/datasets`

The fourth step, data modelling involved dividing the created spectrograms into training, validation and testing datasets in a ratio 70:20:10. The dataset was trained and tested using a sequential CNN model created based on different parameters and layers. The training and testing of the dataset using various pre-trained models like Resnet, Inception and EfficientNet were performed in this stage.

The fifth step, evaluation and result are to evaluate the performance of the trained models based on a classification report with parameters like accuracy, recall, precision and F1-score. Accuracy is the sum of true predicted values in respect to all predicted true and false values i.e. (TP+TN)/(TP+TN+FP+FN). Recall or sensitivity is the ratio of TP / (TP + FN) i.e., correctly predicted languages in comparison to all predicted positive and negative observations for a language. Similarly, precision is the ratio for Precision = TP/ (TP+FP) i.e., correctly predicted positive observations for a particular language in comparison to the total predicted positive observations for the language. F1-score is the weighted average of precision and recall. A confusion matrix is also created to identify the number of audio samples misclassified under different languages.

4 Design Specification

To create a spoken language identification system, a new sequential CNN model is designed to classify the three different languages considered in the experiment. The architecture of the CNN model is defined below

1. Input: The input shape of the spectrogram images passed to the model is (224 X 224 X3). The images are passed in a batch size of 20.

2. Convolutional layer: The model consists of 8 convolutional layers with a filter size of 32, 64 and 128. Each layer has a kernel size of (3 X 3). The padding is set to the same to ensure that the output of the layer has the same dimensions as the input.

3. Activation Layer: ReLu activation function is used to address the issue of vanishing gradients.

4. Regularization: Each convolutional layer is integrated with two L2 regularization functions. The values are set to Kernel Regularizer = .001 and Bias Regularizer = .001to reduce the chance of overfitting.

5. Pooling layers: Four Max pooling layers have been implemented with a kernel size of (2X2).

6. Dropout layers: Five dropout layers have been used in the model. Each parameter is set to 0.5. The layer is used to address overfitting challenges.

7. Flatten Layer: One flatten layer is used to convert the output of convolutional into one single output feature vector.

8. Fully Connected Layer: one fully connected dense layer is used with a filter size of 512.

9. Output Layer: The output layer has three nodes and is equipped with a sigmoid activation function. The model is optimized using Adam optimizer with a learning rate of 0.0001. The loss function used is binary cross-entropy.

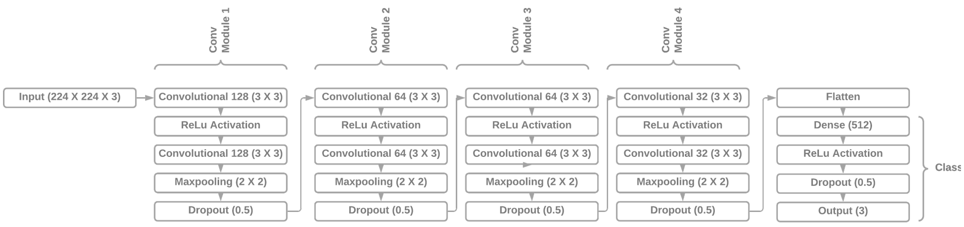

Figure 2: Proposed Model Architecture

5 Implementation

5.1 Setup Google Cloud Platform (GCP) Environment

The created CNN model is an 8 layers model with 2,724,803 trainable parameters. The model is compact enough to be trained and tested using a CPU and the experiments can be done using a google collab notebook. However, the pre-trained models used in the experiments are heavy and training the models using a CPU or a GPU based google collab lead to excessive RAM usage leading to the kernel crash. Also, using google collab, the models could not be trained for extended epochs.

Therefore, based on the conclusions of the literature survey, a GPU based virtual machine was the optimum choice for training the pre-trained models. Google Cloud Platform (GCP) is a cloud-based service provided by Google. The cloud provides a service to create a GPU based virtual machine. The services are chargeable but provide unrestricted access to a virtual machine with GPU capabilities. A machine with an NVIDIA Tesla T4 GPU was created and used for training and testing the models in the experiment. The training or models was faster and smoother as compared to that on a CPU. A Jupyter python notebook is created on the virtual machine and is used for further training and testing models for the experiment.

5.2 Model Implementation

To train and test the CNN and pre-trained models, following implementation steps were followed:

1. The audio files were converted from .mp3 format to .wav format using the pydub python library.

2. The .wav files were converted into mel-spectrograms using the librosa library with hyperparameters for the Mel banks being n_fft=2048, sr=22050, n_mels=128.

3. The mel spectrograms were then renamed and stored in a folder. The folder contained 1000 samples for each of the three languages English, Hindi and Irish. A VBScript file was created to automate the naming functionality. The nomenclature followed is as follows:

   (a) English: English_0001 to English_1000

   (b) Hindi: Hindi_0001 to Hindi_1000

   (c) Irish: Irish_0001 to Irish_1000

The spectrograms were then consolidated in excel, with class 0 assigned to English, 1 assigned to Hindi and 2 assigned to the Irish language.

4. The excel was imported and stored in a dataframe. The dataframe was shuffled to introduce randomization in the training of the languages and dummy variables were created for each of the categories.

5. The folder with the 3000 images is loaded into the python notebook using the ImageDataGenerator functionality and the data augmentation parameters mentioned above are applied to it.

6. The CNN model is compiled and a summary is printed for the new model.

5.3 Training and Testing the CNN Model

The sample size for the training set is 2000 images and the validation size is 600. The batch size for the final implementation is 100 for the training set and 50 for the validation set. While the learning rate with Adam optimizer is set to .0001. Initially, a combination of batch sizes, learning rates and activation functions were used to train the model. The description of combinations is as follows

1. Training batch size = 64, Validation batch size = 32, Learning rate = .001, Activation function(Final Layer) = 'ReLu'

2. Training batch size = 32, Validation batch size = 32 and Learning rate = .001, Activation function(Final Layer) = 'ReLu'

3. Training batch size = 64, Validation batch size = 32 and Learning rate = .0001, Activation function(Final Layer) = 'ReLu'

4. Training batch size = 32, Validation batch size = 32 and Learning rate = .0001, , Activation function(Final Layer) = 'ReLu'

For all the above cases, the accuracy of the model would get stuck as 0.667, for both training and validation datasets. The reason is that the ReLu activation is a linear function and the gradient for all the trainable neurons and outputs becomes stagnant as all individual layers are trained with linear weights. Therefore, a non-linear activation function was used in the output layer. Sigmoid is utilized instead of softmax as sigmoid calculates the probability of each output separately while softmax function interrelates probabilities of all the outputs and sum it to 1. Since the differentiation between each language should be clear, sigmoid serves as an optimum choice.

Same combinations of batch sizes and learning rates are used with the sigmoid activation function. The experiment also performs a multi-label classification instead of a multi-class classification. The goal is to calculate the probability of each language from the output layer and choose the language with maximum probability to finalize the spoken language. Therefore, different to the state-of-the-art models, our experiment employs binary cross-entropy instead of categorical cross-entropy function with a combination of sigmoid activation in the output layer instead of softmax function. The best results are produced with the following parameters: No. of. Epochs = 145, learning rate = .0001, Training batch size = 100, Validation batch size = 50, colour spectrum = RGB.

6 Evaluation

The aim of the experiments is to compare the performance of different models on the created mel spectrograms and determine the best performing model based on accuracy for language identification tasks. A total of 4 experiments were conducted to test the performance of the models.

The evaluation for the CNN model is captured in Section 6.1, Resnet50 in Section 6.2, InceptionV3 in Section 6.3 and EfficientNet in Section 6.4. The accuracy of the different models on the test dataset is shown in Table 1.

Table 1: Accuracy

Models	CNN	Resnet	InceptionV3	EfficientNet-B0
Accuracy	93.50%	89.40%	92.25%	92.75%

6.1 CNN Model

The aim of the experiment is to differentiate between languages using the CNN model. The below result are achieved using following parameters No. of. Epochs = 145, learning rate = .0001, Training batch size = 100, Validation batch size = 50, colour spectrum = RGB parameters.

Table 2: CNN Classification Report

Languages	Precision	Recall	F1-Score
English	0.92	0.88	0.90
Hindi	0.95	0.96	0.95
Irish	0.92	0.96	0.94

The model is generating good recall, precision and F1-score for all three languages as shown in Table 2. The high score for all three parameters is supported by the confusion matrix as shown in Fig.3 where the number of misclassified languages (in the black colour) is minimal when compared to correct predictions made by the model. Also, the misclassification is not too high for any one language indicating that the trained model is performing balanced classification.

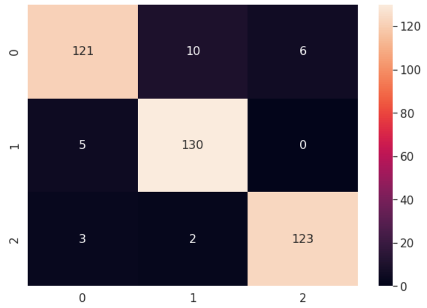

Figure 3: CNN Confusion Matrix

The key performance of the model is indicated by the graph plot as shown in Fig.4 for the accuracy and loss throughout different epochs of the training cycle. The accuracy for both training and validation sets grow together till 100 epochs indicating that the model is training well for both training and validation datasets. After 110 epochs, the training accuracy keeps increasing while the validation accuracy becomes stable indicating that the model has achieved convergence after 110 epochs. The convergence indicates that the model is trained and learned all the features till 110 epochs and very limited feature recognition happen after the point.

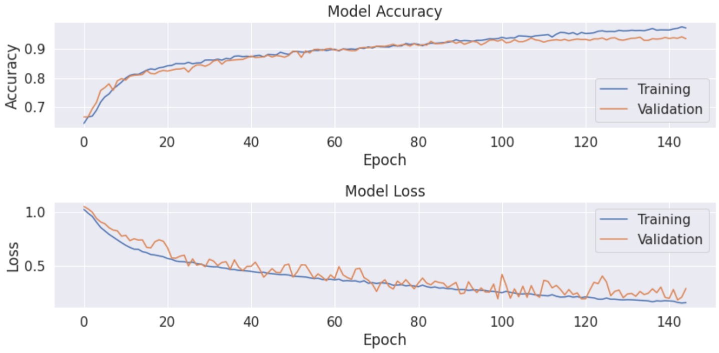

Figure 4: CNN Accuracy and Loss Plot

Also, plotting the loss function indicates that throughout training, the loss value for the training dataset decreases constantly but the loss value for the validation dataset fluctuates rapidly. Although, it constantly decreases and is throughout in close proximity to the training loss indicating minor overfitting of the model. However, for over 90% of the epoch cycle validation loss is comparatively equal to the above training loss, which indicates a good model, only 10% of epochs have high fluctuations in the validation loss. This happens due to the limited data points for training and smaller batch size. With a larger dataset and computation power, batch size can be increased. Similarly, the validation loss is not lesser than the training loss indicating that the model is not underfitting.

6.2 Resnet50

The below result are achieved using following parameters No. of. Epochs = 50, learning rate = .001, Training batch size = 20, Validation batch size = 20, colour spectrum = RGB parameters.

Table 3: ResNet50 Classification Report

Languages	Precision	Recall	F1-Score
English	0.83	0.87	0.85
Hindi	0.92	0.89	0.91
Irish	0.93	0.92	0.92

The recall, precision and F1-score are good for the Irish language, the scores are only satisfactory for the other two languages, indicating that the model is able more accurately identify Irish as compared to the other two languages.

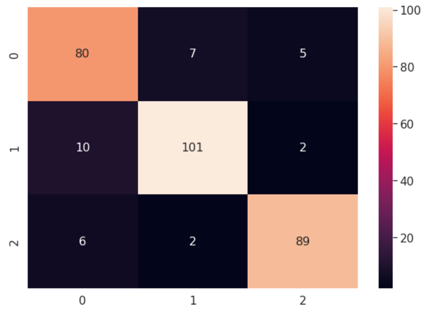

Figure 5: Resnet50 Confusion Matrix

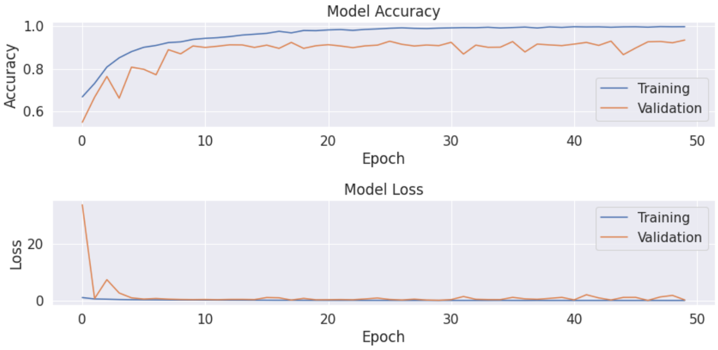

Figure 6: Resnet50 Accuracy and Loss Plot

The model is initially trained with a batch size of 100 and 50 similar to the CNN model, however, the accuracy of the model severely drops to 80% with the sample size. When training the model with an equal batch size of 20, the accuracy of the model increases however, the training dataset trains quickly but the validation accuracy increases gradually. The validation accuracy is marginally less than the training accuracy. However, since the losses decrease continually the model does not exhibit overfitting but the model.

6.3 InceptionV3

The below result are achieved using following parameters No. of. Epochs = 50, learning rate = .001, Training batch size = 50, Validation batch size = 50, colour spectrum = RGB parameters.

Table 4: InceptionV3 Classification Report

Languages	Precision	Recall	F1-Score
English	0.81	0.98	0.89
Hindi	0.98	0.93	0.96
Irish	1.00	0.86	0.93

The recall and precision values for InceptionV3 are opposite to the results generated using the Resnet model. The recall value for English is high however the precision value goes low indicating that a higher number of False positives for the language because 21 samples amongst all the English samples were classified as Irish as indicated by cross-validation of the confusion matrix. Similarly, for Irish, the precision value is high while the recall value is low again which indicates a high number of false negatives.

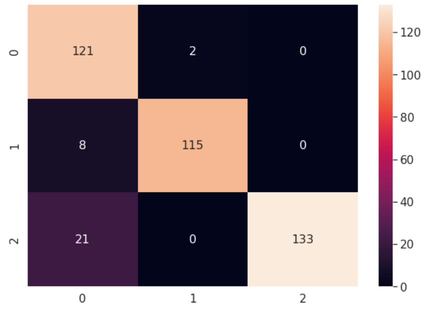

Figure 7: InceptionV3 Confusion Matrix

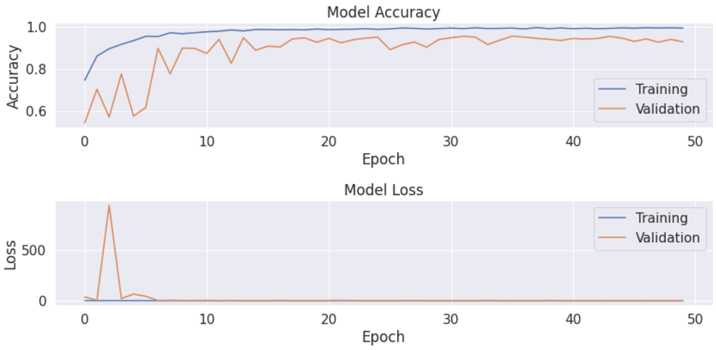

Figure 8: InceptionV3 Accuracy and Loss Plot

For the model accuracy, the model reaches 99% training accuracy quickly. However, similar to the Resnet model the validation accuracy is not at par with it. The loss is consistent with the ResNet model. Both the curves observe a single spike at the beginning for validation loss but since the loss reduces constantly and always has a higher value than the training loss, overfitting of the model can be ruled out. Also, both the losses are in close proximity and constantly decrease removing the possibility of underfitting.

6.4 EfficientNet

The below result are achieved using following parameters No. of. Epochs = 50, learning rate = .001, Training batch size = 20, Validation batch size = 20, colour spectrum = RGB parameters.

Table 5: EfficientNet Classification Report

Languages	Precision	Recall	F1-Score
English	0.88	0.93	0.90
Hindi	0.98	0.90	0.94
Irish	0.94	0.95	0.94

The recall and precision values for the model are promising and is consistent with the expected values. No particular languages are highly misclassified and indicate good identification and classification for the test dataset. However, examining the loss curve, the validation loss is consistently higher than the training loss indicating that the model is overfitting indicating that the model might produce bias in predictions when tested on a different dataset.

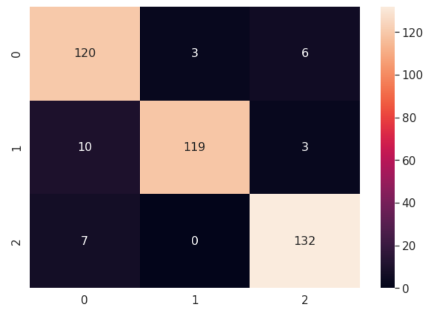

Figure 9: EfficientNet Confusion Matrix

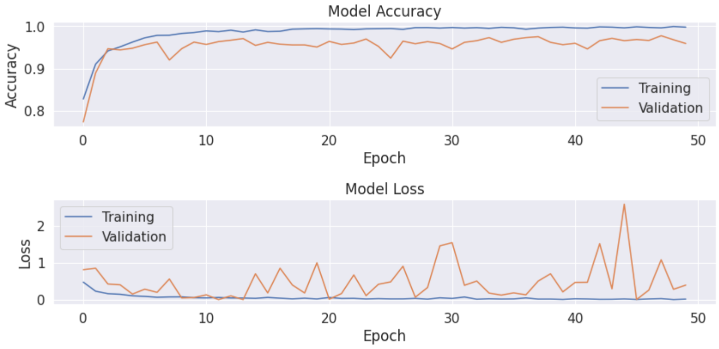

Figure 10: EfficientNet Accuracy and Loss Plot

6.5 Discussion

The CNN model is efficient, provides higher accuracy on the test set and can perform better classification than the three pre-trained models. The model also trains well as indicated by the accuracy plot since both the training and validation accuracy are consistent with each other. The model learns the weights to identify features from the training process and performs balanced classification for all three languages. The captured loss observes quite ups and downs along with slight overfitting at the beginning of the training process, however, the loss values constantly decrease as the training progresses. The best way to address the challenge faced by the model while capturing loss is by using a larger dataset and a larger batch size for training and testing datasets. However, to establish that more computation power and resources will be required.

The pre-trained models trained too quickly on the training dataset however, the model trains slowly for the validation dataset indicating that the models take time to identify the weights for feature extraction. EfficientNet observes overfitting, therefore, achieves good results for the trained datasets but the performance might not be replicated for a new dataset. Both Resnet and InceptionV3 perform well for the Hindi language, however, the results for both the models are contradicting for English and Irish languages.

7 Conclusion and Future Work

The aim of the research was to differentiate amongst languages with same linguistic background from the Indo-European Language Family. The research proposes a CNN based language identification model that can help differentiate between spoken languages using mel spectrograms for feature extraction. Results demonstrate that the proposed CNN model can differentiate between the Irish, English and Hindi languages with an accuracy of 93.50%. A limitation of the research is limited computation power. With more computation power, the model can be trained on larger audio samples with larger batch sizes for training and testing datasets.

Resnet50 and InceptionV3 perform well for Hindi language and highest number of misclassified samples are for English and Irish languages. The EfficientNet model needs to be trained with larger sample sizes to achieve better performance as there was overfitting using the limited dataset. Furthermore, accent of the speakers in the dataset also plays a crucial role in building a language identification system. As part of future work, the models can also be trained with different accents considering the demography of the speakers contributing to the dataset.

The work can also be potentially improved by training the proposed model using a larger dataset and optimizing the training process using additional image augmentation techniques. With image augmentation, the size of the dataset increases and requires more computational power. Transfer learning can also be applied using the pre-trained models such as DenseNet, NASNET and MobileNet to improve the accuracy of the language identification system. .

References

Arla, L. R., Bonthu, S. and Dayal, A. (2020). Multiclass spoken language identification for indian languages using deep learning, Institute of Electrical and Electronics Engineers Inc., pp. 42–45.

Bartz, C., Herold, T., Yang, H. and Meinel, C. (2017). Language identification using deep convolutional recurrent neural networks.

Basodi, S., Ji, C., Zhang, H. and Pan, Y. (2020). Gradient amplification: An efficient way to train deep neural networks, *Big Data Mining and Analytics* **3**(3): 196–207.

Bianco, S., Cadène, R., Celona, L. and Napoletano, P. (2018). Benchmark analysis of representative deep neural network architectures.

Bohra, N. and Bhatnagar, V. (2021). Language identification using stacked convolutional neural network (scnn), Institute of Electrical and Electronics Engineers Inc., pp. 20–25.

Celano, G. G. A. (2021). A resnet-50-based convolutional neural network model for language id identification from speech recordings.

China, C., Bisharad, D. and Laskar, R. H. (2018). Automatic classification of indian languages into tonal and non-tonal categories using cascade convolutional neural network (cnn)-long short-term memory (lstm) recurrent neural networks, *2018 International Conference on Signal Processing and Communications (SPCOM)*, pp. 492–496.

Demircan, S. and Örnek, H. K. (2020). Comparison of the effects of mel coefficients and spectrogram images via deep learning in emotion classification, *Traitement du Signal* **37**: 51–57.

Dörfler, M., Bammer, R. and Grill, T. (2017). Inside the spectrogram: Convolutional neural networks in audio processing, *2017 International Conference on Sampling Theory and Applications (SampTA)*, pp. 152–155.

Hall, J., O'Quinn, W. and Haddad, R. J. (2019). An efficient visual-based method for classifying instrumental audio using deep learning, *2019 SoutheastCon*, pp. 1–4.

Heracleous, P., Takai, K., Yasuda, K., Mohammad, Y. and Yoneyama, A. (2018). Comparative study on spoken language identification based on deep learning.

Kaiyr, A., Kadyrov, S. and Bogdanchikov, A. (2021). Automatic language identification from spectorgam images, Institute of Electrical and Electronics Engineers Inc.

Lopez-Moreno, I., Gonzalez-Dominguez, J., Plchot, O., Martinez, D., Gonzalez-Rodriguez, J. and Moreno, P. (2014). Automatic language identification using deep neural networks, Institute of Electrical and Electronics Engineers Inc., pp. 5337–5341.

Lozano-Diez, A., Plchot, O., Matějka, P. and Gonzalez-Rodriguez, J. (2018). Dnn based embeddings for language recognition, Vol. 2018-April, Institute of Electrical and Electronics Engineers Inc., pp. 5184–5188.

Lu, Q., Li, Y., Qin, Z., Liu, X. and Xie, Y. (2020). Speech recognition using efficientnet, ICST, pp. 64–68.

Meghanani, A., S., A. C. and Ramakrishnan, A. G. (2021). An exploration of log-mel spectrogram and mfcc features for alzheimer's dementia recognition from spontaneous speech, Institute of Electrical and Electronics Engineers Inc., pp. 670–677.

Purwins, H., Li, B., Virtanen, T., Schlüter, J., Chang, S. Y. and Sainath, T. (2019). Deep learning for audio signal processing, *IEEE Journal on Selected Topics in Signal Processing* **13**: 206–219.

Revay, S. and Teschke, M. (2019). Multiclass language identification using deep learning on spectral images of audio signals.

Shintri, R. G. and Bhatia, S. K. (2015). Analysis of mfcc and multitaper mfcc feature extraction methods.

Singh, G., Sharma, S., Kumar, V., Kaur, M., Baz, M. and Masud, M. (2021). Spoken language identification using deep learning, *Computational Intelligence and Neuroscience* .

Stolar, M., Lech, M., Bolia, R. S. and Skinner, M. (2018). Acoustic characteristics of emotional speech using spectrogram image classification, *2018 12th International Conference on Signal Processing and Communication Systems (ICSPCS)*, pp. 1–5.