Classifying Music Genres | A Comparison of Multilayer Perceptrons and Support Vector Machines Using Convolved Mel-Spectrograms

Brenner Swenson
Brenner.Swenson@city.ac.uk

Abstract

Deep learning techniques have been proven to classify images with high accuracy, and the same methodologies can be applied to audio tracks converted to spectrograms for use with image classifying methods. In this work, the multilayer perceptron and support vector machine algorithms are compared for their ability to classify the genres of a large music database using features from convolved Mel-spectrograms; the analysis yields that both algorithms achieve comparable results.

1. Introduction

Music is a vital part of human culture; music can define cultural movements, bring a crowd to tears, or spark a political revolution. Music genres are loosely defined as systems of orientations, expectations, and conventions that bind together an industry, performers, critics, and fans in making what they identify as a distinctive sort of music. [1] Genres allow musicians and fans of music to communicate their tastes and provide recommendations to others based on the genres they like.

Over the past decade consumers have switched from listening to music via CDs or even explicitly purchasing albums on a platform like iTunes to relying primarily on subscription services like Spotify or Apple Music to provide content recommendations based on their tastes and user behaviour. As of December 2019, Spotify provided over 50 million tracks on their platform [2]. It would be virtually impossible to manually classify each of these tracks as belonging to a certain genre (let alone possible sub-genres); thus, machine learning techniques can be implemented to automatically perform the classification task.

The field of music information retrieval (MIR) focuses on the extraction of meaningful data from audio and can be used to manipulate, catalogue, and even write music. There are generally two MIR approaches for processing music for classification with machine learning models: 1. extracting features from the raw audio files such as a song's tempo, tonnetz, spectral contrast, etc. for use with most machine learning models or 2. converting the audio files to a frequency spectrogram for techniques commonly associated with image classification e.g. convolutional neural networks.

This paper focuses on the latter of the above approaches. Using feature maps from convolutional layers, this paper aims to compare, contrast, and criticize the support vector machine and the multilayer perceptron algorithms as the classifying output layer in a convolutional neural network.

1.1 Multilayer Perceptron

The multilayer perceptron is a supervised machine learning model and was one of the first artificial neural networks. The multilayer perceptron (MLP) is, as it sounds, multiple layered perceptron algorithms stacked next to one another. The perceptron algorithm uses labelled input features to iteratively update the weights of connected neurons using an error function that is calculated at the end of the network; as the weights of each neuron connection are updated via backpropagation, the network attempts to slowly (depending on a learning rate) approximate a function to classify or predict the target value. The network ultimately defines a mapping that learns the parameters resulting in the best function approximation.

Pros	Cons
• Input data types are very flexible (no feature derivation) • Highly tuneable and fast training time • Able to approximate any continuous function with finite number of neurons and a single hidden layer [3]	• Prone to overfitting without use of regularisation techniques • Often converge to local minima • Difficult to interpret results • Results are often not reproducible

1.2 Support Vector Machine

Another supervised learning method, the support vector machine, similar to logistic regression, is perpetuated by the linear function $\boldsymbol{w}^T\boldsymbol{x} + b$ attempting to draw a line between classes. SVMs do not output probabilities akin to logistic regression, they instead output the predicted class with 100% certainty. Most importantly, SVMs attempt to draw the best separating line where logistic regression does not.

SVMs are able to do this in infinite dimensions by means of maximum margin hyperplanes constructed via the kernel trick. As not all datasets are linearly separable, the SVM's learning algorithm effectively replaces the $\boldsymbol{x}$ in $\boldsymbol{w}^T\boldsymbol{x} + b$ with the output of some **kernel function**. These kernel functions allow the algorithm to learn non-linear models in n-dimensional space; additionally, the kernel functions are often much more computationally efficient than simply computing the dot product between the input vector and feature weights. Although initially SVMs were introduced for binary classification problems, they have since been extended to multiclass problems.

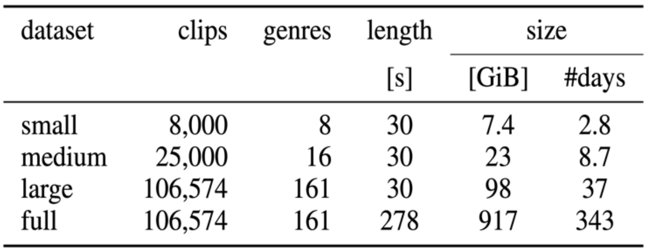
Pros	Cons
• Much fewer hyperparameters • Guaranteed global optimum • Relatively memory efficient • Very effective in high dimensions • Performs well with small datasets	• Longer training times • Fails to perform well with overlapping classes (noise points) • Do not output probability estimates • Difficulty choosing best kernel

1.3 Hypothesis Statement

It is expected that the SVM will greatly outperform the MLP as the output layer of a CNN due to its superior ability to generalise when provided with high-dimensional data via the kernel trick. However, MLPs can be highly tuned to a specific problem via hyperparameter optimisation that could prove to outperform the SVM.

2. Dataset

This paper utilises the Free Music Archive (FMA) dataset, a publicly available dataset containing a total of 106,754 audio tracks, pre-computed features, and metadata. [4] The data is an export of the FMA, an open and free music collection organized by WFMU, an independent freeform radio broadcasting station in New Jersey. Defferrard et al. introduce the FMA dataset as a solution to many MIR problems that require a large comprehensive dataset. Additionally, as seen in Table 1, they propose size-varied subsets as of the FMA for researchers who may be constrained by limited computational resources.



dataset	clips	genres	length	size	
			[s]	[GiB]	#days
small	8,000	8	30	7.4	2.8
medium	25,000	16	30	23	8.7
large	106,574	161	30	98	37
full	106,574	161	278	917	343

Table 1 – FMA-proposed data subsets

This paper analyses the raw audio of the small subset. Containing 8,000 unique tracks across 8 different genres, the small subset's genre labels are balanced evenly with 1,000 tracks belonging to each of the 8 top genres. Each of the subsets also contains a training (80%), validation (10%), and testing (10%) split to ensure that research on the FMA is reproducible. To ensure that individual artists do not contaminate the testing or validation data and support generalisation, one artist's tracks can only belong to one of these three splits.

3. Methods

3.1. Methodology

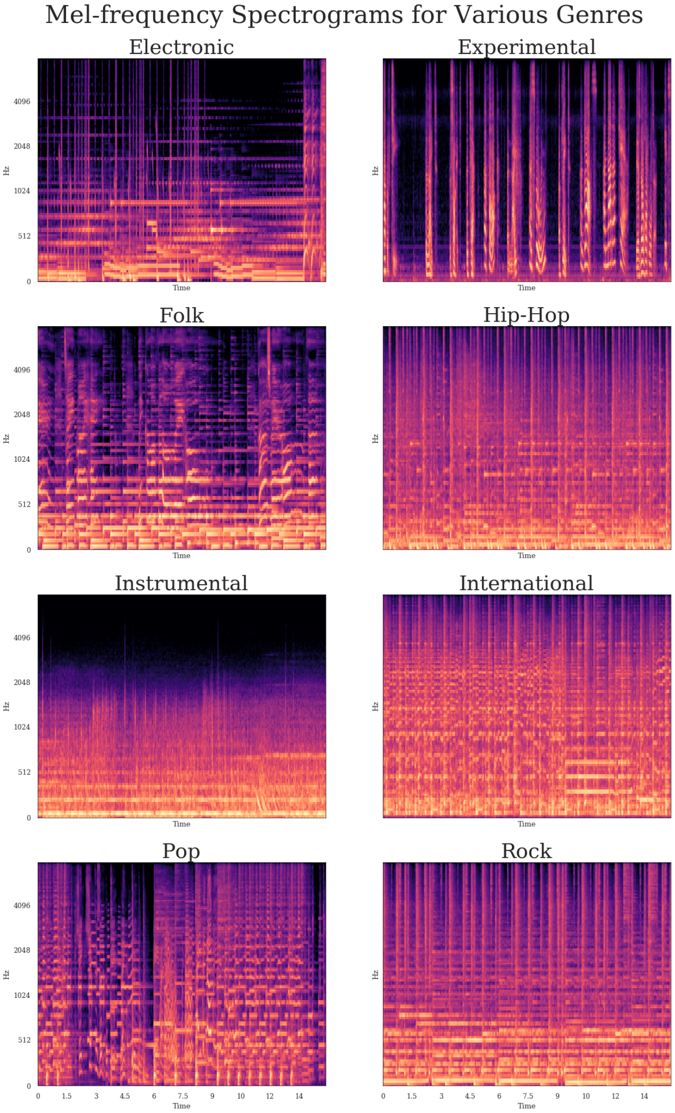


Figure 1. Mel-frequency spectrograms of sampled songs in the FMA dataset

With the advent of deep learning algorithms, hand-crafted features are no longer required for music genre classification. Convolutional neural networks (CNN) are able to detect objects and patterns in images invariant of the object's location; they are able to do this by identifying edges, such as a person's mouth or eyebrows, in an image and associates those features with the training label provided to the algorithm.

As such, a piece of music or audio can be converted to a spectrogram that can essentially be treated as an image for classification purposes. Over the last few years, **Mel-spectrograms** have become the most important feature set used in speech and audio processing. [5] A spectrogram is a 2D depiction of an audio signal, with the frequency on the vertical axis and time on the horizontal axis. An example of Mel-frequency spectrograms can be seen in Figure 1; the colouring indicates where the decibels at various frequencies are higher/lower.

There exist intuitive qualitative differences between each genre 's spectrograms, for instance the Hip-Hop example displays periodic spikes at the same frequencies over what appears to be an equal interval. This can be interpreted as the beat of the song; whereas with folk music, there is much less uniformity and regularity in the example. These differences are what allow the CNN to create features and what ultimately allows algorithms to predict which song belongs to which genre. Using the LibROSA Python package [6], spectrograms were created for each song using a **FFT** window of 2048 and a **hop length** of 512. 15 seconds of each 30 second clip were then utilised.

3.2 Feature Extraction CNN Architecture

The implemented CNN architecture is very similar to and inspired by the architecture used by Y. Yu, S. Luo and S. Liu et al [7] in their hybrid CNN-RNN attention-based classifier. Table 2 contains the architecture in question. The spectrogram is fed to 5 blocks of comprised of a convolutional layer, a **max pooling** layer, with batch normalization layer between each one.

Table 2. CNN Architecture for Feature Extraction

Layer Type	Filter shape	Stride and padding	Input shape
Conv + ReLU	3 x 3	1 x 1, same	128 x 512 x 1
MaxPooling	2 x 2	2 x 2, -	128 x 512 x 16
BatchNorm	-	-	64 x 256 x 16
Conv + ReLU	3 x 3	1 x 1, same	64 x 256 x 16
MaxPooling	2 x 2	2 x 2, -	64 x 256 x 32
BatchNorm	-	-	32 x 128 x 32
Conv + ReLU	3 x 3	1 x 1, same	32 x 128 x 32
MaxPooling	2 x 2	2 x 2, -	32 x 128 x 64
BatchNorm	-	-	16 x 64 x 64
Conv + ReLU	3 x 3	1 x 1, same	16 x 64 x 64
MaxPooling	4 x 4	4 x 4, -	16 x 64 x 128
BatchNorm	-	-	4 x 16 x 128
Conv + ReLU	3 x 3	1 x 1, same	4 x 16 x 128
MaxPooling	4 x 4	4 x 4, -	4 x 16 x 64
BatchNorm	-	-	1 x 4 x 64
Flatten	-	-	1 x 4 x 64
Dense	-	-	1 x 256

Batch normalization (BN) scales the activations between each convolution block between 0 and 1, which speeds up training times. The addition of BN also lessens the amount of hidden unit covariance shift which improves generalisation and can allow for the use of higher learning rates.

The smaller filter and **stride** shapes in the first few layers are designed to extract highly granular features that are then down sampled via max pooling in the last two layers with larger kernels and strides. This technique promotes the learning of more robust features which help in the prevention of overfitting. [7]

3.3 Training, Parameters, and Evaluation

The training and evaluation process utilised the FMA pre-defined 80%, 10%, 10% split for training, validation, and testing data, respectively. To prevent overfitting and introduce variance to the training data, each training sample (with an original array width of 640) is randomly cropped along the horizontal axis to achieve a width of 512 seen in the CNN input layer. Random cropping, an image augmentation method, prevents the model from memorising the data as a result of how it was cropped for training. In the context of music, it makes logical sense as songs often begin with silence, and the model should not expect there to be silence at the beginning of each sample. With random cropping, the training sample can be sliced to begin anywhere in the first 3 seconds of the song. To further normalise the data, the spectrogram values were scaled according to the minimum and maximum values in each array, thus bounding each value between 0 and 1. This normalisation allows for quicker data processing by a GPU or CPU.

3.3.1 Training, Parameters, and Evaluation – MLP

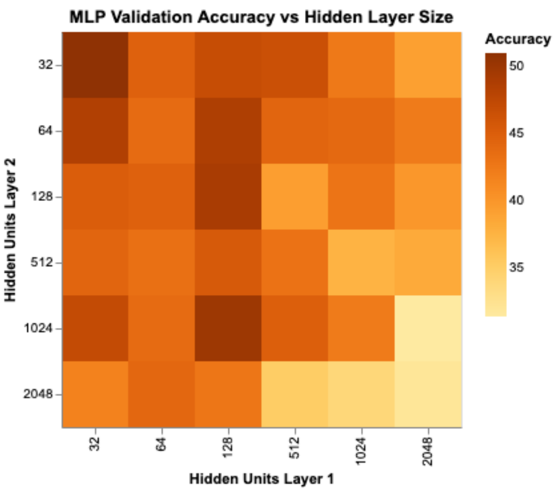


Figure 2. Hidden Layer Grid Search Results

Determining the number of layers for the MLP and the number of neurons in each layer was somewhat of a challenging process as the time to train the model fully was quite cumbersome given the input data sizes and the convolutions required. Given the universal approximation theorem and supported by Senac et al. in their similar architecture choices [8], it seemed fitting to choose two fully connected layers prior to the **softmax** classification layer with differing numbers of hidden units. To narrow down the search for optimal hidden units and determine the MLP's sensitivity to number of neurons, a grid search experiment across 6 possible values for each layer was implemented. It would take drastically too long to train each iteration of the grid search on the full dataset, so this experiment was conducted on a smaller subset of the FMA dataset where 1500 samples were randomly chosen.

This experiment roughly took 12 hours to run in its entirety; and the results were rather surprising. As seen in Figure 2, the larger the number of neurons in both layers, the worse the MLP performed across the experiment's 15 **epochs**.

Qualitatively it can be observed that the smaller MLP configurations performed better, but this could also be due to the fact that they require less time to train, and can learn quicker, hence achieving better results in a shorter amount of time in comparison to their more complex counterparts.

During the training of the MLP, the ADAM (Adaptive Moment Estimation) optimiser was used with a linearly decreasing learning rate initially set at 0.01. ADAM is attractive for this task as it requires very little memory and is able discover optimal learning rates for its own hyperparameters. [9] To avoid overfitting and reduce generalisation error, an L2 weight penalty of 0.01 was added to encourage the network to use smaller weights. In the same vein, a **minibatch** size of 128 was used, representing only 2% of the entire training set. A smaller minibatch size allows for a significantly smaller memory footprint during training, and also improves generalisation performance. [10] The training data was shuffled at the beginning of each epoch.

3.3.2 Training, Parameters, and Evaluation – SVM

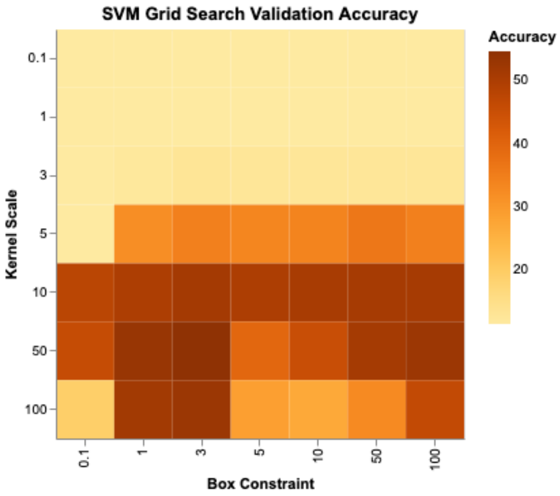


Figure 3. Box Constraint and Kernel Scale Grid Search Results

The training process for the SVM was much simpler for the SVM in comparison to the MLP as there are very few hyperparameters. With the input data in 256 dimensions, a Gaussian kernel was used as it is especially robust against the curse of dimensionality. To investigate the SVM's sensitivity to certain hyper-parameters, a similar grid search experiment was conducted to that of the MLP, but instead of hidden layer size, box constraint and kernel scale were tested. It should be noted that these values were tested using standardized input features.

For comparison, the SVM grid search was also performed on a subset of 1500 samples. For generalisation purposes, each grid search iteration utilised 10-fold cross validation, with the best candidate's accuracy being returned and plotted in Figure 3. It can be observed that the models' performance is very sensitive to the **kernel scale** values, but the **box constraint**, i.e. the penalty on misclassifications, is not as impactful on performance.

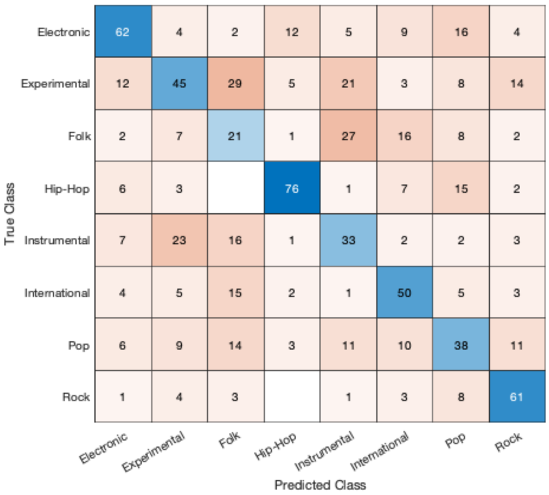


Figure 4. SVM Confusion Matrix

4. Results, Findings

Grid search was used for the MLP as a way to filter for best hyperparameter ranges, the best accuracy on the completely unseen test set (different from the grid search validation data) was observed to be 47.13% using 128 and 64 neurons for the first and second layer, respectively. For the SVM, a Bayesian optimisation approach was applied to the kernel scale and box constraint values. This technique allowed for high optimisation and performed slightly better than the MLP with 48.25% accuracy using the optimised values of 1 and 15.39 for the box constraint and kernel scale, respectively.

Surprisingly, the SVM did not outperform the MLP by a very large margin as initially hypothesised. The SVM did train much quicker than the MLP, which is desirable for practical applications; this is likely because SVM's training times aren't as susceptible to high-dimensional data when compared to the MLP. SVMs are able to train without the use of backpropagation, instead finding optimal separating hyperplanes, and can generalise to unseen much better which could explain the higher testing accuracy. SVM variants are also often preferable in an online-learning scenario where new data is introduced frequently as they are very computationally efficient [11]; in the music domain, e.g. millions of new songs being uploaded to a streaming service, this could be highly desirable.

As seen in the SVM's confusion matrix in Figure 4, it is reassuring to see that the model misclassifies points that a human could possibly misclassify. For example, Experimental vs. Instrumental could largely contain the same characteristics. Also, of note is the complete absence of any misclassifications for the Hip-Hop genre; as mentioned in the commentary on Figure 1, Hip-Hop is highly recognisable due to the consistent rhythmic features associated with the genre. When compared to the MLP's genre misclassifications, they are largely the same except for Experimental being confused with Rock, which may highlight a possible inability in the MLP to differentiate between certain features. The SVM's misclassification of Folk and Experimental and the MLP's misclassifications of Experimental and Rock genres both warrant further investigation.

5. Conclusion, Lessons Learned

This analysis compared the support vector machine and the multilayer perceptron as music genre classification algorithms using convolved Mel-spectrograms as input data. It yields that both methodologies are comparable for the task at hand, as the SVM does not outperform the MLP by a material margin. It is reassuring that both algorithms tend to misclassify tracks that are often confused by humans, prompting the exploration of other popular MIR datasets to determine if the confusion relates only to the FMA data and not a property of the learning techniques. The accuracy figures are comparable to the baselines reported by FMA [11] utilising similar methodologies.

This paper did not fully explore all the possible network architecture configurations due to limited computational resources. The existing models in this paper could also be expanded to the larger subsets of the FMA dataset as the learning may have been constrained by only 8 genres and 8,000 tracks. Future work could explore the exploration of recurrent networks and the inclusion of a song's temporal features as explored by Chillara et al. and their successful use of **LSTM**'s in conjunction with convolved spectrograms to achieve high accuracy on the FMA dataset. [11]

6. References

[1] J. C. Lena and R. A. Peterson, 'Classification as Culture: Types and Trajectories of Music Genres', *Am. Sociol. Rev.*, vol. 73, no. 5, pp. 697–718, 2008.

[2] 'Spotify — Company Info', *Spotify*. https://newsroom.spotify.com/company-info/ (accessed Apr. 05, 2020).

[3] A. Kratsios, 'The Universal Approximation Property: Characterizations, Existence, and a Canonical Topology for Deep-Learning', *ArXiv191003344 Cs Math Stat*, Feb. 2020, Accessed: Apr. 10, 2020. [Online]. Available: http://arxiv.org/abs/1910.03344.

[4] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, 'FMA: A Dataset For Music Analysis', *ArXiv161201840 Cs*, Sep. 2017, Accessed: Apr. 06, 2020. [Online]. Available: http://arxiv.org/abs/1612.01840.

[5] M. Dörfler, T. Grill, R. Bammer, and A. Flexer, 'Basic filters for convolutional neural networks applied to music: Training or design?', *Neural Comput. Appl.*, vol. 32, no. 4, pp. 941–954, Feb. 2020, doi: 10.1007/s00521-018-3704-x.

[6] 'LibROSA — librosa 0.7.2 documentation'. https://librosa.github.io/librosa/ (accessed Apr. 11, 2020).

[7] Y. Yu, S. Luo, S. Liu, H. Qiao, Y. Liu, and L. Feng, 'Deep attention based music genre classification', *Neurocomputing*, vol. 372, pp. 84–91, Jan. 2020, doi: 10.1016/j.neucom.2019.09.054.

[8] 'Music Feature Maps with Convolutional Neural Networks for Music Genre Classification | Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing'. https://0-dl-acm-org.wam.city.ac.uk/doi/abs/10.1145/3095713.3095733 (accessed Apr. 09, 2020).

[9] J. Duchi, E. Hazan, and Y. Singer, 'Adaptive Subgradient Methods for Online Learning and Stochastic Optimization', p. 39.

[10] D. Masters and C. Luschi, 'Revisiting Small Batch Training for Deep Neural Networks', *ArXiv180407612 Cs Stat*, Apr. 2018, Accessed: Apr. 12, 2020. [Online]. Available: http://arxiv.org/abs/1804.07612.

[11] A. Bordes, S. Ertekin, J. Weston, and L. Bottou, 'Fast Kernel Classifiers with Online and Active Learning'. JMLR.org, Dec. 01, 2005.

[12] S. Chillara, 'Music Genre Classification using Machine Learning Algorithms: A comparison', vol. 06, no. 05, p. 8, 2019.

7. Glossary

Box constraint – A SVM hyperparameter used to penalise misclassified points (often referred to as C). The larger the box constraint value, a higher cost (misclassification penalty) is applied.
Convolution – The act of passing a kernel matrix over an input matrix thus creating a feature map that is ingested by the next layer.
Epoch – A full cycle of the training data during a learning algorithm's training process.
FFT – Fast Fourier transform; an algorithm that computes the discrete Fourier transform of a sequence, thus converting the input data from the time domain to a frequency domain.
Hop length – A spectrogram hyperparameter used to determine the number of samples between successive frames, e.g. the columns in the spectrogram.
Kernel function – A mathematical function that transforms input data to a specified form used in the SVM algorithm. Kernels return the inner product of two points in an acceptable feature space.
Kernel scale – Often referred to as 'gamma' when used with a Gaussian kernel SVM, the kernel scale defines the influence of a single training example's reach.
LSTM – Long short-term memory, a recurrent type of artificial neural network with feedback connections for sequence learning.
Max pooling – The down sampling process that reduces the dimensionality of a feature map, thus promoting faster training times and avoiding overfitting by promoting the maximum value of the discretized pool shape.
Mel-spectrograms – 2D representations of audio computed via the FFT on windowed segments of the input signal. The frequency values are binned using the Mel scale, a unit of pitch that normalizes frequencies according to how the human ear perceives them.
Minibatch – A small sample of training data used to calculate model gradient and error, thus updating the model's weights.
Softmax – An activation function that outputs a probability between zero and one for each class in a multi-class model.
Stride – A parameter that decides how far the convolution/pooling filters shift around the input matrix when performing their respective actions.

8. Implementation Details

The FMA tracks were pre-processed in Python utilising the LibROSA [6] package to create the spectrograms which were then saved down as .mat files for consumption by Matlab. The matrices were required to be saved into directories with their genre labels as the directory's titles for use with Matlab's ImageDataStore objects; this file manipulation was done using Python as well. All below implementation instructions are intended for Matlab.

To run the saved models on test data, download all of the necessary Matlab files and data from Google Drive https://drive.google.com/open?id=1TC7DYFdRaJDVVwEDrF43EcxbtaYm5YfL. The Matlab files should be ran in the same working directory as the 'data' folder.

For the MLP's training process, 30 epochs were used, and dropout layers were added between each of the fully connected layers and the output layer. The file that trained the best MLP classifier is titled 'coursework_mlp_train.m'. The grid search used to create the graph in Figure 2 can be found in the file titled 'coursework_mlp_train_gridsearch.m'. To achieve the reported test accuracy values for the MLP, run the file 'coursework_mlp_test.m'.

For the SVM, the training file titled 'coursework_svm_train' contains a commented block of code relating to the Bayesian optimisation process used to ultimately determine the hyperparameters; it is included for illustrative purposes and not intended to be ran. The SVM is able to use the CNN feature maps by extracting them from the 5th pooling layer. It should be noted that for both the MLP and SVM, the validation and test data are converted to augmentedImageDatastore objects, but the data is not actually augmented. They are cropped in the centre to ensure normality when testing. The grid search used to create Figure 3 can be found in the 'coursework_svm_train_gridsearch.m' file. To obtain the reported accuracy results for the SVM, run the file titled 'coursework_svm_test.m'.