FOMO AI

A Computer Vision and Deep Learning Project
to multi-classify genres for nightlife music posters

Professor:
Daniele Pannone

Class:
Computer Vision & NLP

Developers:

Ganni Galea Curmi	matr. 2005657
Giovanni Montaruli	matr. 1988069
Davide Cerci	matr. 1996149

Idea Stemming

Fomo:

What prompted the initial idea of how we as a group were going to define our machine learning project was a real life necessity of one of this group's members. He recently co-founded an app-based start-up called Fomo, of which the aim is to connect users with event organisers, essentially allowing people to discover new events in the area.

Most of the above mentioned events are relevant to music (nightlife, discos, concerts, locals, bars, live music, etc.) and the company needed a streamlined way to correctly identify each of the hundreds of characteristics, such as the places, the music played, the artists, and more, for each of the events that every day are going through the Fomo's app. Most of the events are currently scraped, thus correctly classifying each event into its correct genre is essential.

In order to make Fomo's life easier, our idea was to simplify at least one of these hunts for event characteristics by providing an AI model that could accurately assign one or more genres of music to the events' posters just by observing the artwork. In this way, each event could immediately have a label assigned with their genres, without having to do it by hand, one by one.

In order to train the model, though, we needed a big enough database of music-related artworks. Thus, our idea was to use Spotify Music albums, already labelled by genre, and see whether this can correctly predict the music event posters genre. (Dataset - over 21k labelled album covers)

Spotify API:

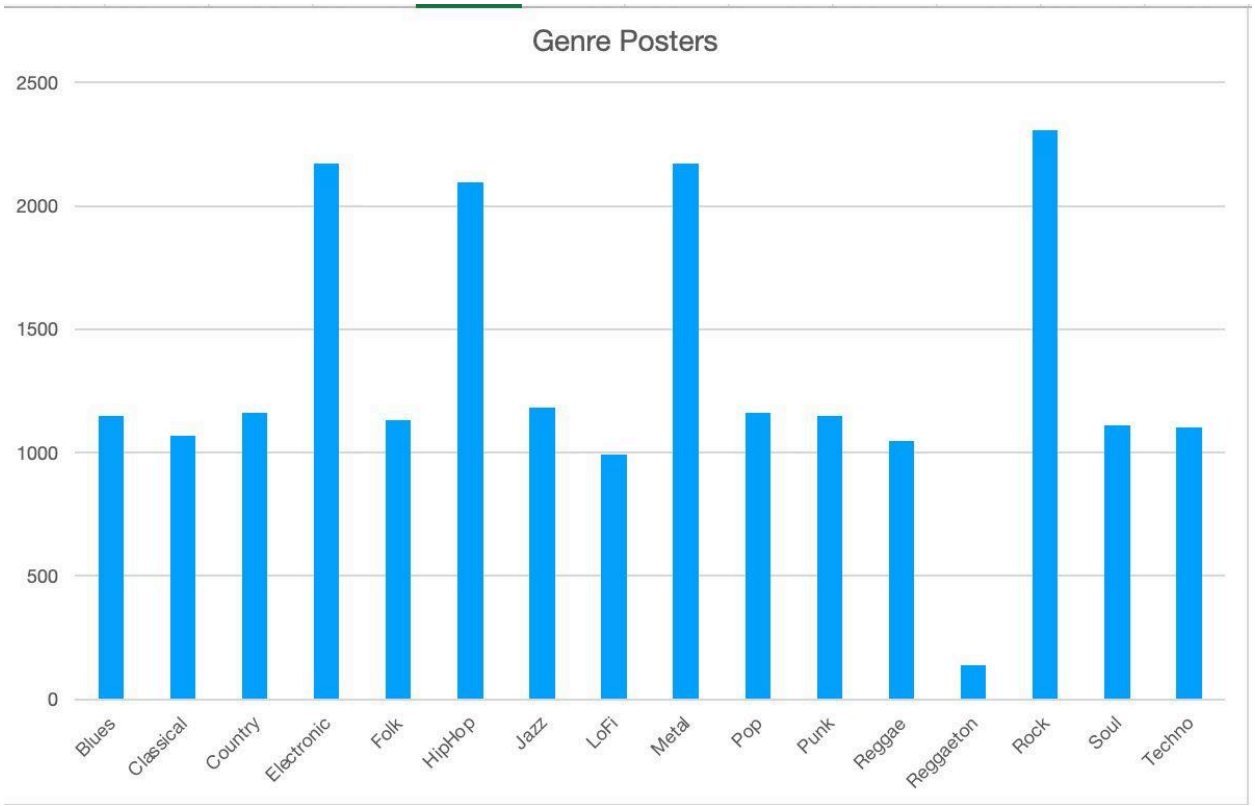
Similarly to the MNIST database, where an enormous number of handwritten numbers are available to be used, the music industry has huge databases around the internet that allow us to have a large dataset to work with. We ended up choosing Spotify because the API gave us access to tens of thousands of readily available album covers for dozens of different genres. Moreover, the album covers were well labelled (for our supervised learning approach) and thus we just had to format them in the same way, for easier image processing.

At the end of the day, the objective was to build a multi-label image classifier trained on Spotify's Album Covers and use that trained knowledge to accurately guess the genre(s) of any Fomo poster we wish to submit. Each submitted poster must have at least one genre assigned, together with the confidence level of the model's guess.

Notes:

A fundamental aspect of the code was of course the actual Neural Network that we trained. It is important to note that we went through several different strategies and attempts in order to achieve our objective and, in the beginning, our aim was to create our own Convolutional Neural Network. However, this approach resulted in a CNN that was too simple to accurately perceive the huge number of variables that we were looking to analyse. Our simple CNN wasn't able to learn properly as it always ended up predicting the most common labels in our dataset. To avoid a complete do-over, we decided to try to use VGG16 with weights trained from imagenet preloaded into the convolutional blocks, randomly initialised fcc layers, and the last layer resized to be of size 16, the total number of genres from our dataset.

Another important aspect to consider is the non-uniformity of our dataset. We tried our best to provide as many samples of one specific label as any other one in our dataset. However, some musical genres are simply less popular or less numerous than others, so we expected a slight bias on predictions towards the most popular ones. As it can be seen while running a test, this can be true for some of the genres (like Reggaeton) where sometimes they happen to be predicted less. Moreover, we chose to combine some similar genres to try and counter the imperfect nature of the database, something that may have added more bias towards some bigger and "colourful" genres (like Electronic, which has House and Drum 'n' Bass within it).



Distribution of genres

Another important thing to consider is that, despite them usually having attributes in common, music covers can range incredibly widely when it comes to their artworks and stylistic choices compared to their genre. Sometimes it's hard for us humans to guess a genre correctly based on the artwork alone, we can imagine how challenging it can be for a machine. The interesting part, however, is that sometimes our model can predict more accurately than us humans, meaning that it must pick up some subtle attributes that we wouldn't. On the other hand, however, the opposite is also true: some obvious guesses for humans are very confusing for the machine. Here, an integration of NLP could be a strong solution to some wrong classifications.

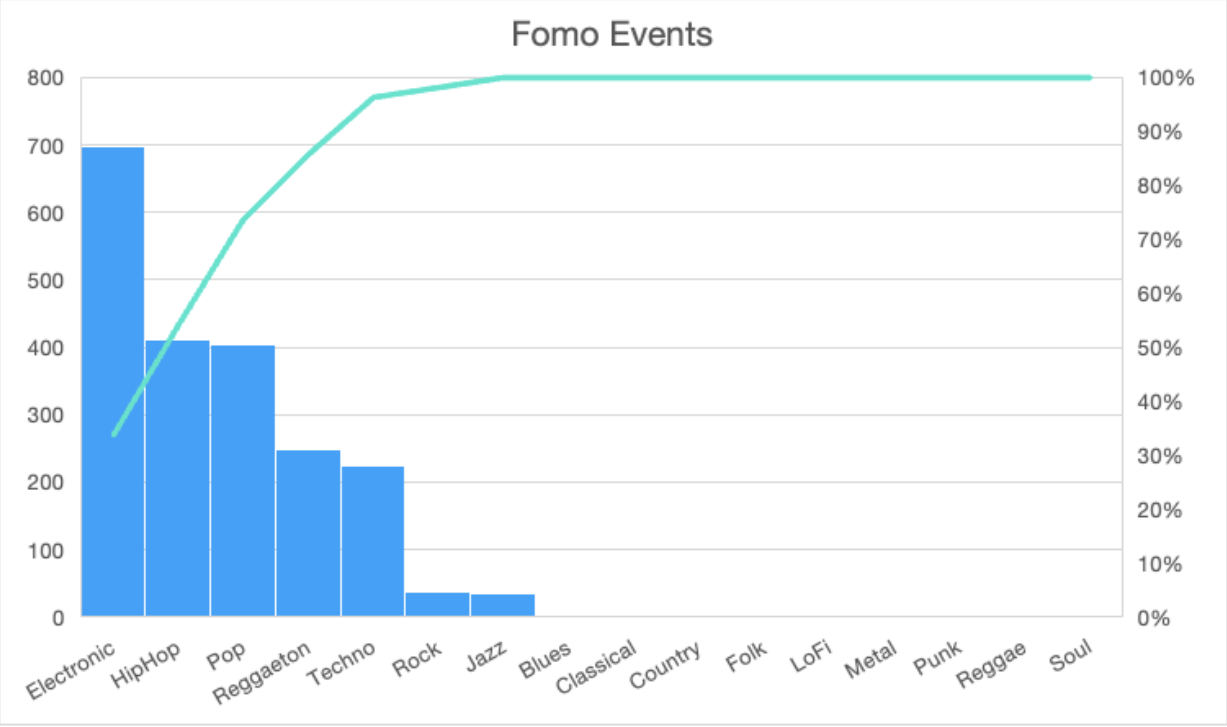
Album Covers Sample

Fomo Posters Sample

Prior to testing, the fomo images were first encoded into the same genres as the model was trained on. This meant that some of the 'Fomo Genres' were combined into a similar genre.

```
csv_to_model_genre_map = {
    "disco": "Electronic",
    "jazz": "Jazz",
    "retro": "Pop",
    "rap": "HipHop",
    "hip-hop": "HipHop",
    "reggaeton": "Reggaeton",
    "edm": "Electronic",
    "r&b": "Soul",
    "house": "Electronic",
    "industrial techno": "Techno",
    "trance": "Techno",
    "acid": "Techno",
    "psychadelic trance": "Techno",
    "deep house": "Techno",
    "rock": "Rock",
    "commercial": "Pop",
    "tech house": "Techno",
    "melodic techno": "Techno",
    "techno": "Techno",
}
```

The mapping of genres for fomo dataset



The distribution of the fomo dataset

Implementation

Here it follows a broad-view explanation on the workflow of our code.

Definition of Constants and Parameters:
Set up essential parameters such as thresholds, batch size, number of epochs, learning rate, and paths to dataset and CSV files.

Custom Dataset Loader Setup:
We created a MultiLabelDataset class to handle loading of a multi-label dataset from a CSV file and a directory containing images.
We implement methods to read images and their corresponding labels, apply transformations, and return them for training.

Model Architecture Definition:
Here we define a base model class (ImageClassificationBase) to outline common training steps and logging.
We create a specific model class (MusicPosterClassification) that adapts the pre-trained VGG16 model for the task, modifying the classifier to suit the multi-label classification of music posters.

Data Preparation:
We are loading the dataset using the custom dataset loader, applying necessary transformations (resizing and tensor conversion).
Next we are splitting the dataset into training and validation sets to ensure model evaluation during training.

Training and Validation Data Loaders:
Here we initialise data loaders for both training and validation datasets to manage data batching and loading during training.

Evaluation Function Definition:
We implement a function (evaluate) to assess model performance on the validation dataset, calculating metrics such as precision, recall, and F1 score.

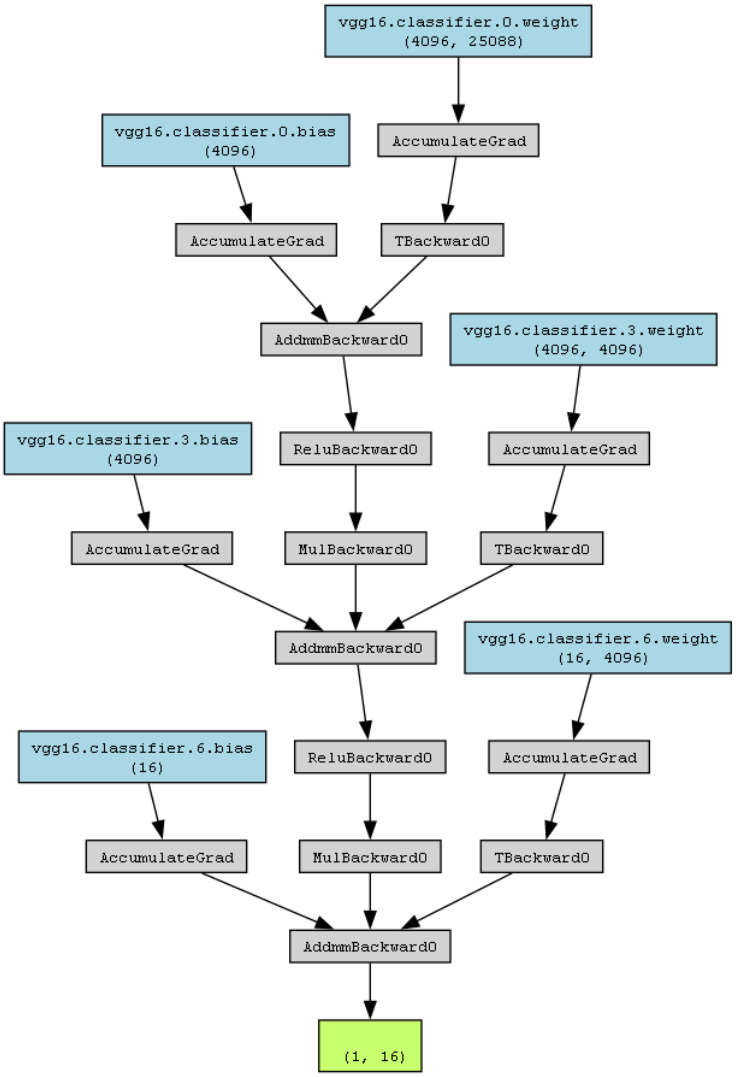
Training Function Implementation:
After defining a fit function to train the model over a specified number of epochs, including backpropagation, optimization steps, and logging we incorporate early stopping and model checkpointing based on validation loss to prevent overfitting and ensure model robustness. In all of our attempts, we got to early stopping by the 9th (out of 30) epoch, since apparently the model wasn't getting any better training after that. (after going through over 160k images)

Model Training Execution:
Prepare the model and initiate training using the defined fit function.
Log training metrics for monitoring and evaluation purposes.

Subset Dataset Creation:
This was optional but also a crucial step to quickly test our model, saving us time and computation resources. Before the main training starts, we create a subset of the dataset using stratified sampling to maintain label distribution (hence ensuring a balanced representation of classes).

Model Evaluation and Prediction:
Post-training, we are using the trained model to predict labels for new images (specifically Fomo Music Posters), processing the images and applying the model to obtain predictions.
We match the model's binary predictions with class names and confidence levels, providing meaningful output for Fomo end users.

Model Diagram:



Results Overview

The assessment of our classifier's performance was conducted on a set of 846 event posters from Fomo, with the requisite preprocessing to align them with the model's trained genre categories. In consideration of the high number of multi-labeled posters and the skewed genre representation within the Fomo dataset, the analysis of our model's results was conducted with particular attention to the particular distribution of the dataset's composition.

The key performance metrics were as follows:

- **"At Least One" Accuracy:** This primary metric shows the model's strength, with an 83% success rate in identifying at least one accurate genre per poster.
- **Precision:** Although a figure of 25.4% may seem low, it is indicative of the model's selective accuracy amidst the complexity of multi-label classification.
- **Recall:** At 64.2%, the model demonstrates a solid ability to detect pertinent genres from the dataset.
- **F1 Score:** The score of 0.32 reflects the equilibrium between precision and recall, which is particularly notable given the diversity of the dataset.
- **Top 1 Accuracy**: At 41%, and **Top 2 Accuracy:** at 54.6%, these metrics defined as most confident genre predictions, either singularly or in pairs, affirmed that the model's highest-confidence predictions are accurate roughly half of the time.

In the sphere of music and event visuals, marked by high variability and artistic expression, album covers are inherently inconsistent. This inconsistency was anticipated to escalate when transposing learned features to event posters, yet the model's performance was still promising.

The model's ability to discern genres from the eclectic visual languages of event posters—akin to a task quite formidable for human observers—indicates a successful application of the learned features across different domains also thanks to our application of the VGG16 model for our purposes.

While the metrics on their own may not appear striking, they are significant in light of the model's ability to approach the classification of event posters—a task with a higher degree of difficulty due to the less standardised visual cues as compared to album covers.

The results illustrate a proof of concept that holds promise for cross-media genre classification, suggesting that the model's capabilities could be further refined for practical application within the Fomo app.

Challenges

At first, our setup was only capable of identifying a single genre per image, but we quickly realised the need to upgrade our approach to handle multiple genres simultaneously. This required a significant adjustment to our dataset, incorporating one-hot encoding to accommodate multi-label classification. Impressively, the model adapted to multi-label scenarios better than anticipated, even though it was initially trained on single-label data.

Our initial attempt with a basic custom CNN proved to not be complex enough. The model was too simplistic, often defaulting to the most numerous genres in the dataset and struggling with accurate genre recognition due to its high classification threshold. This was particularly challenging given the visual diversity within music genres, which can sometimes be ambiguous, even to the human eye.

We also faced a learning curve in selecting appropriate metrics for multi-label classification, with Recall, F1 Score, and Binary Cross Entropy Logloss, etc. These metrics provided clearer insights into the model's performance across various genres.

Given the computational demands of processing colour images, we initially considered leveraging Microsoft Azure Compute's free credits for access to more powerful computing resources. However, the practical challenges of this approach led us to start with a smaller dataset. This strategy allowed us to iteratively refine our model, gaining valuable insights before applying it to the entire dataset.

Overall, the project was a balancing act between adapting our model for multi-label classification, fine-tuning its architecture, and navigating computational constraints. By starting small and scaling our efforts gradually, we developed a model capable of discerning the nuanced visual signatures of different music genres, marking a successful resolution to our initial challenges.

Potential Improvements

The following are some improvements that could be implemented in our model both in general and for the specific purposes of the company we built it for.

NLP:
For text recognition within the image, we can implement techniques like Optical Character Recognition (OCR) to extract text from the image. Once the text is extracted, NLP can be used to process and understand the context or sentiment of the text. This additional textual information can help the model make more accurate predictions, perhaps by improving classification accuracy, by considering both visual and textual cues within the album covers.

In the context of the Fomo company, most event posters will have text that is attached to the image as a description or metadata. In this case NLP can be used to analyse and extract meaningful insights from this textual data. NLP techniques can help in understanding the intent, sentiment, or context of the poster. It's important to note that, just like the image alone, the description or the tags of an event might not have all the information to correctly classify the poster. Hence a combination of the two sources of information is the best way to get a prediction as close to reality as possible. For instance the description of the poster can provide valuable keywords that improve the relevance of search results.

In both cases, the combination of image features and textual information processed through NLP can enhance the overall performance of our machine learning model by providing a more comprehensive understanding of the image content. This, in turn, leads to more accurate and context-aware predictions. Thus fine tuning the model further before applying it in the wild will result in much more accurate classification for Fomo's goal.