Relatório Final

Aprendizagem Computacional II

Trabalho realizado por:

Diogo Mendes nº202108102
Gonçalo Brochado nº202106090
Tiago Coelho nº202105004

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

U.PORTO
FC FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

1

Breve descrição do problema de classificação

Este projeto tem como principal objetivo o desenvolvimento de um classificador de sons urbanos, provenientes de um dataset fornecido, nomeadamente “urbansound8k”, usando deep learning.

O dataset é constituído por 8732 sons que estão categorizados em 10 classes distintas, nomeadamente:

1. Ar condicionado
2. Buzina de um carro
3. Crianças a brincar
4. Ladrar de um cão
5. Perfuração
6. Motor de um veículo
7. Tiro
8. Martelo pneumático
9. Sirenes
10. Música de rua

A tarefa em questão é bastante desafiadora pois as dez classes abrangem uma ampla gama de sons urbanos. Devido a possíveis ruídos urbanos e/ou baixa qualidade do som, todo este processo irá envolver um pré-processamento dos respetivos dados.

Já na parte dos modelos de classificação é pretendido que se implemente dois dos seguintes modelos:

- Um classificador baseado num neurónio multicamada (MLP)
- Um classificador baseado numa rede neuronal convolucional (CNN)
- Um classificador baseado numa rede neuronal recorrente (RNN)

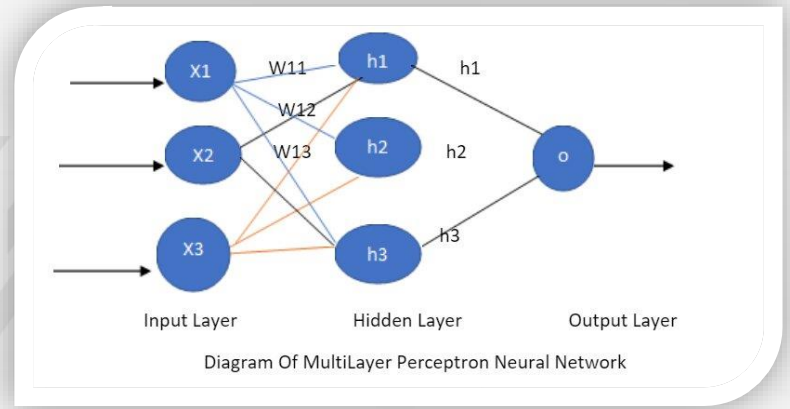
Resumindo, iremos implementar 2 algoritmos de deep learning para tentar obter o melhor classificador possível para este dataset, realizando um pré-processamento dos dados para cada algoritmo e aplicando estratégias ideais para cada um deles de modo que, no final, seja possível realizarmos uma avaliação das implementações e dos resultados.

2

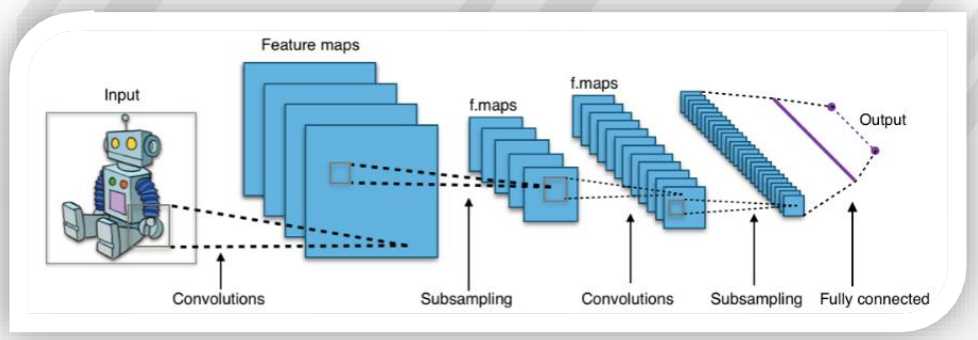
Breve descrição das soluções deep learning consideradas

Para este trabalho decidimos que iriamos utilizar como algoritmos de deep learning um classificador baseado num neurónio multicamada (MLP) e numa rede neuronal convolucional (CNN)

O modelo MLP é uma Rede Neuronal na qual todos os nós estão interconectados com os nós de diferentes camadas, formando uma rede densamente conectada. A camada de entrada recebe os dados e cada neurónio nessa camada representa uma característica específica. Os neurónios nas camadas ocultas realizam transformações não lineares nos dados, aprendendo representações mais complexas e abstratas. Finalmente, a camada de saída gera as previsões ou classificações.



Já o modelo CNN é uma classe de redes neuronais profundas que são usadas principalmente para analisar dados que têm uma estrutura espacial ou temporal, como imagens, sons, vídeos, textos, etc. CNN usa uma técnica especial chamada convolução, que consiste em aplicar filtros (também chamados de kernels) sobre os dados de entrada para extrair características relevantes. A convolução é repetida várias vezes, formando camadas ocultas que aprendem representações cada vez mais complexas e abstratas dos dados. A última camada de uma CNN é geralmente uma camada totalmente conectada, que recebe as características extraídas pelas camadas anteriores e realiza a classificação final dos dados em diferentes categorias.



3

Descrição da implementação

No início do projeto, estabelecemos uma linha de base essencial utilizando exclusivamente o primeiro conjunto de dados (folder) de uma base mais extensa. Este passo inicial foi crucial para entender e delinear a abordagem a adotar ao longo do desenvolvimento do projeto.

A primeira etapa foi a padronização dos áudios. Utilizamos o processo de resample para ajustar todos os áudios para uma frequência comum de 44100 Hz, garantindo uniformidade na representação sonora. Além disso, adotamos uma estratégia de normalização da duração dos áudios, fixando todos em 4 segundos. Isso foi alcançado através da aplicação de zero padding nos áudios que eram originalmente mais curtos, assegurando uma consistência temporal no conjunto de dados.

Em seguida, procedemos à extração de características essenciais para a análise acústica, nomeadamente os Coeficientes Cepstrais de Frequência Mel (MFCCs). Optamos por extrair 15 desses coeficientes para cada áudio, considerando 321 janelas de tempo. Cada janela representa um segmento do áudio ao longo do tempo, permitindo capturar variações acústicas significativas.

Começamos por aplicar o classificador Multilayer Perceptron (MLP), onde optamos por utilizar as médias desses coeficientes como representação sintetizada de cada áudio.

No que diz respeito ao modelo de classificação, adotamos uma abordagem simples com uma camada de entrada e uma camada de saída utilizando a função softmax para classificação multiclasses. A ativação ReLU foi utilizada na camada de entrada. O treino foi realizado por 15 epochs, e a otimização foi obtida com o otimizador Adam.

A avaliação do modelo revelou uma accuracy significativa no conjunto de teste. Para analisar mais a fundo o desempenho, empregamos gráficos elucidativos. A matriz de confusão proporcionou uma visão detalhada da precisão do modelo em cada classe, enquanto os gráficos de evolução da perda e accuracy ao longo das epochs forneceram insights sobre a aprendizagem do modelo.

Para além disso aplicamos também uma abordagem mais avançada utilizando o classificador Convolucional Neural Network (CNN) com duas dimensões. Este modelo profundo consistiu em camadas convolucionais para deteção de padrões e redução de dimensionalidade, seguidas por camadas densas para a classificação final. A ativação 'ReLU' foi empregue para introduzir não-linearidades.

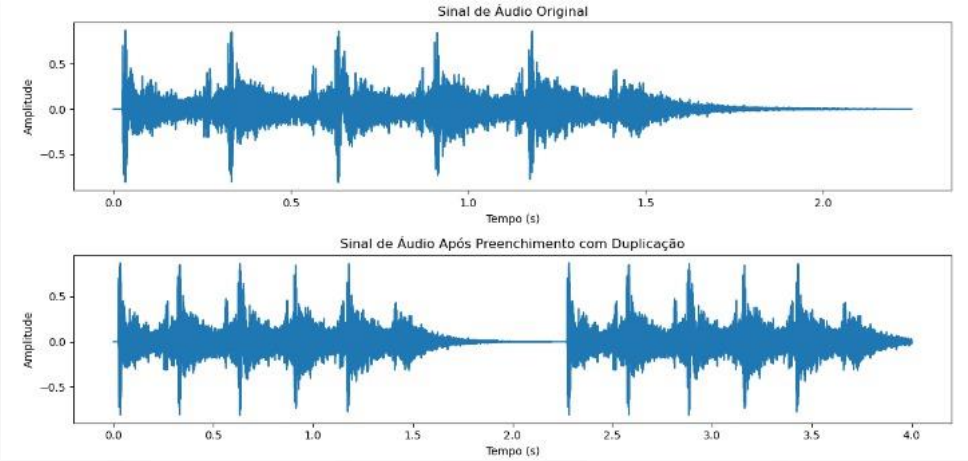
A avaliação do modelo revelou uma accuracy significativa, indicando a sua capacidade para aprender padrões complexos nos dados acústicos. Para uma análise mais aprofundada, empregamos ferramentas visuais, como a matriz de confusão, que proporciona insights sobre a precisão em cada classe, e gráficos para observar a evolução da perda e accuracy ao longo das epochs.

A matriz de confusão destacou a eficácia do modelo em diferentes classes, enquanto os gráficos forneceram uma visão temporal do processo de aprendizagem. Essas representações visuais enriqueceram a análise do seu desempenho, facilitando interpretações mais aprofundadas.

Após a construção da nossa linha de base, exploramos alternativas para aprimorar esse processo e otimizar a representação dos dados acústicos. Nesta fase, concentramo-nos especificamente em técnicas de manipulação do tempo, como a repetição do som e a interpolação.

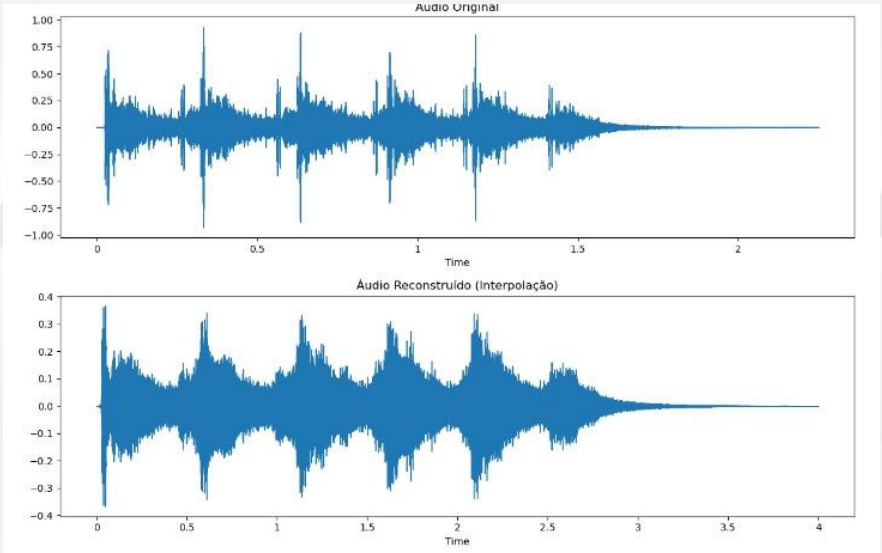
Repetição do Som:

A técnica de repetição do som envolve a duplicação ou multiplicação de um segmento de áudio, que resulta numa extensão temporal do som. Essa abordagem é eficaz para preencher espaços vazios nos áudios que são alvos do zero padding, mantendo a integridade do conteúdo sonoro. A repetição do som permite que o modelo capture mais informações temporais, principalmente em casos de áudios mais curtos, contribuindo para uma representação mais rica e detalhada.



Interpolação:

A interpolação, por sua vez, é uma técnica que visa estimar valores intermédios entre pontos conhecidos. No contexto do processamento de áudio, a interpolação temporal é aplicada para estender ou comprimir a duração de um áudio, ajustando-o para atender a requisitos específicos, como os 4 segundos estabelecidos para nossa linha de base. Ao contrário do zero padding, a interpolação preserva as características temporais do áudio, evitando a introdução de artefactos que podem ocorrer com a simples adição de zeros.



Ao incorporar essas técnicas de manipulação temporal à nossa abordagem de pré-processamento, procuramos mitigar limitações associadas ao zero padding, proporcionando alternativas mais dinâmicas e adaptáveis à diversidade de durações encontradas nos áudios urbanos. Essa otimização visa aprimorar a qualidade da representação dos dados, potencialmente contribuindo para um desempenho mais robusto e generalizado do modelo de classificação de sons urbanos.

Após a implementação das alternativas, conduzimos uma análise comparativa para determinar qual estratégia oferecia melhor desempenho em termos de representação de áudios urbanos. Este processo foi crucial para aprimorar a qualidade da nossa base de dados e, consequentemente, impactar positivamente o desempenho do modelo de classificação. Através da análise comparativa entre as duas técnicas para modelos iguais conseguimos perceber, com a ajuda de métricas como a accuracy, que a interpolação teve uma melhor performance e então decidimos adotar apenas esta estratégia para o decorrer do projeto.

Com isto, começamos novamente por extrair as features, mas desta vez do dataset inteiro com a particularidade de manter a divisão dos 10 folders após a extração das features. Para além disso no caso do modelo classificativo MLP decidimos retirar mais estatísticas tal como o desvio padrão, mínimo e máximo, entre outras, para melhorar o desempenho do modelo.

Para o treino dos modelos usamos crossvalidation onde um dos folders foi usado como dados de teste, outro como dados de validação e todo o resto como dados de treino. Após testarmos com os modelos iniciais, rapidamente percebemos que eram insuficientes e que precisavam de ser melhorados. Ao melhorar os nossos modelos ficamos com o seguinte:

Em termos de parâmetros ajustamos os valores de dropout de l2 regularization e de learning rate assim como a arquitetura dos modelos de forma a prevenir um overfit do modelo (accuracy de treino muito superior a accuracy de validação), usando assim para o classificador:

CNN:

Modelo:

Camadas Convolucionais (Conv2D): O modelo incorpora duas camadas convolucionais com ativação ReLU, destinadas a extrair características relevantes dos coeficientes MFCCs.

Normalização (BatchNormalization): Para acelerar o treino e melhorar a estabilidade, foram introduzidas camadas de normalização em lotes.

Camadas de Pooling (MaxPooling2D): A fim de reduzir a dimensionalidade e preservar características cruciais, camadas de pooling foram estrategicamente adicionadas.

Dropout: Camadas de Dropout foram implementadas para mitigar o risco de overfitting durante o treino.

Regularização (L2): Uma regularização L2 foi aplicada nas camadas convolucionais para controlar a complexidade do modelo e prevenir overfitting.

Estratégia de Treino:

Validação Cruzada (10 Folds): A técnica de validação cruzada foi empregue para avaliar o modelo em diferentes conjuntos de treino e teste, garantindo uma avaliação robusta e generalizável.

Class Weight Balancing: Peso das classes foi balanceado para lidar com desequilíbrios nas classes e melhorar a capacidade do modelo de aprender características de classes minoritárias.

Normalização dos Dados: Os dados foram normalizados para garantir que todas as features estivessem na mesma escala, facilitando o treino e melhorando a convergência do modelo.

Early Stopping: Estratégias como Early Stopping foram implementadas para evitar overfitting e melhorar a eficiência do treino.

Visualização da accuracy por Fold: Foi gerado um gráfico que apresenta a accuracy obtida em cada fold, proporcionando uma visão visual do desempenho do modelo em diferentes conjuntos de dados de teste.

MLP:

Modelo:

O modelo foi concebido como uma rede neural sequencial, composta por camadas densas. Inicialmente, uma camada densa de 512 neurônios, ativada pela função ReLU, seguida por normalização em lote (Batch Normalization) e uma camada de dropout para regularização (evitando overfitting). Este padrão repetiu-se com camadas subsequentes de 256 e 128 neurónios. A camada de saída utilizou a função de ativação softmax para realizar a classificação multiclasse.

Estratégias de Treino:

Balanceamento de Peso das Classes: Foi empregue o balanceamento de peso das classes para lidar com a disparidade de amostras entre diferentes classes, permitindo que a rede neural atribua mais importância às classes menos representadas.

Normalização dos Dados: Os dados de entrada foram normalizados para garantir consistência e convergência mais eficiente durante o treino.

Parada Antecipada (Early Stopping): A técnica de parada antecipada foi implementada para interromper o treino caso não haja melhorias significativas nas métricas de validação, evitando assim o overfitting.

Avaliação e Visualização:

Validação Cruzada (10 Folds): Adotou-se a estratégia de validação cruzada, dividindo os dados em 10 folds distintos para avaliar o desempenho do modelo em diferentes conjuntos de treino e teste.

Gráfico de Acurácia por Fold: Um gráfico foi gerado para visualizar a variação da accuracy em cada fold, fornecendo insights sobre a consistência do modelo em diferentes subconjuntos de dados.

Média e Desvio Padrão: Calculou-se a média e o desvio padrão das accuracies obtidas nos diferentes folds, oferecendo uma visão estatística abrangente do desempenho médio e da variabilidade do modelo.

Tendo em conta os modelos foram ajustados os parâmetros de forma a obter a melhor performance possível, tendo sempre cuidado para evitar overfit, obter uma loss de validação parecida à de treino e uma accuracy de validação próxima da de treino.

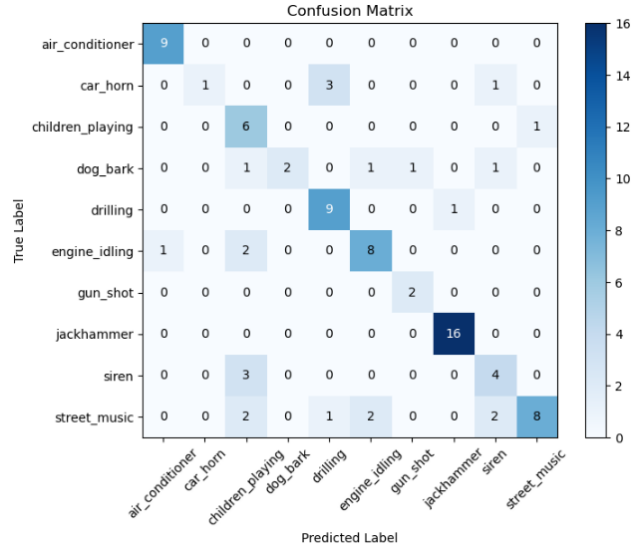
4

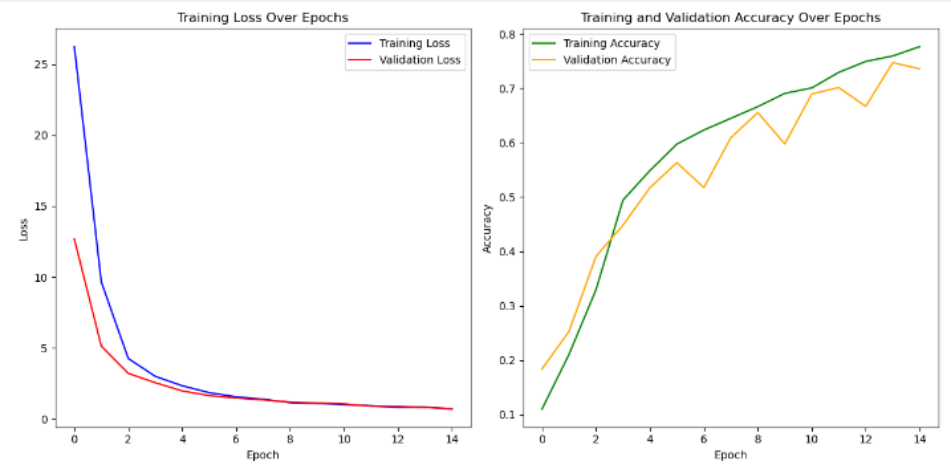
Resultados

Classificação da performance obtida através da implementação de diferentes classifiers.

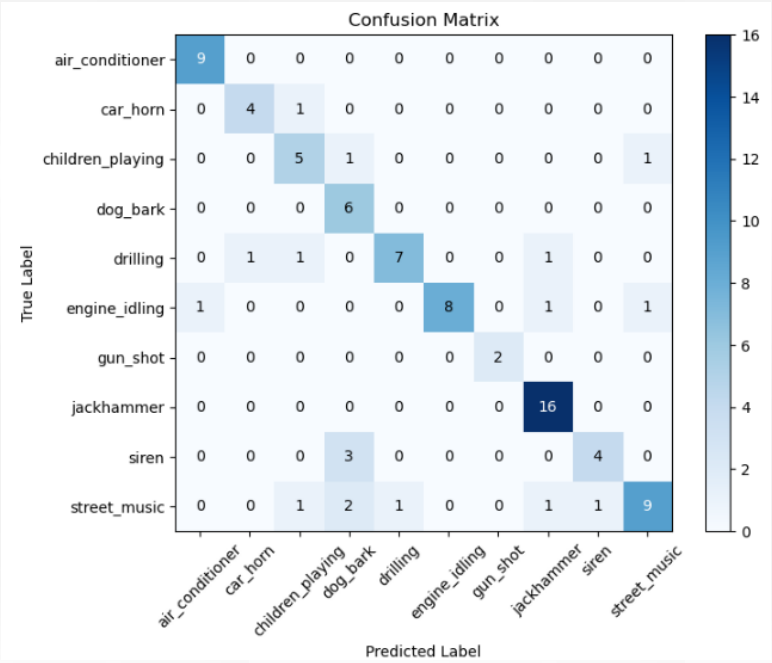
Resultados da base line:

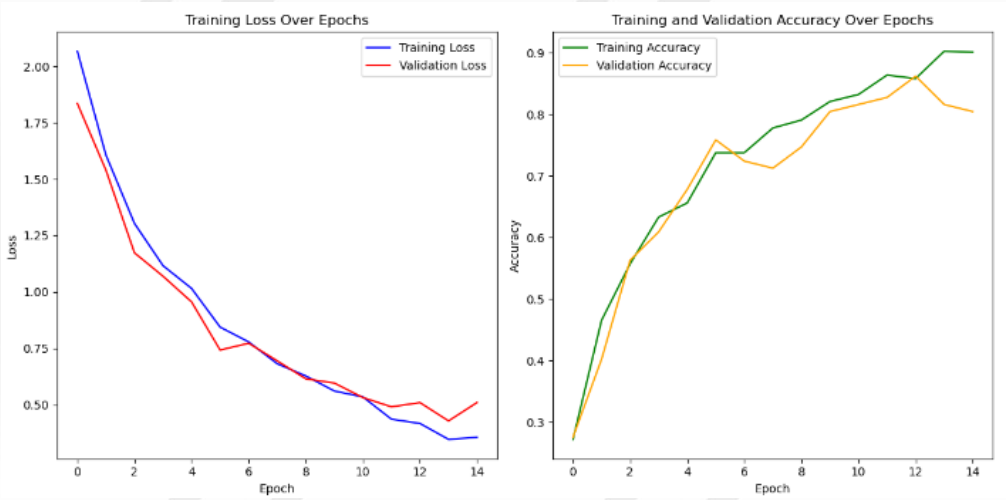
MLP: `Test Accuracy: 0.7386363744735718`





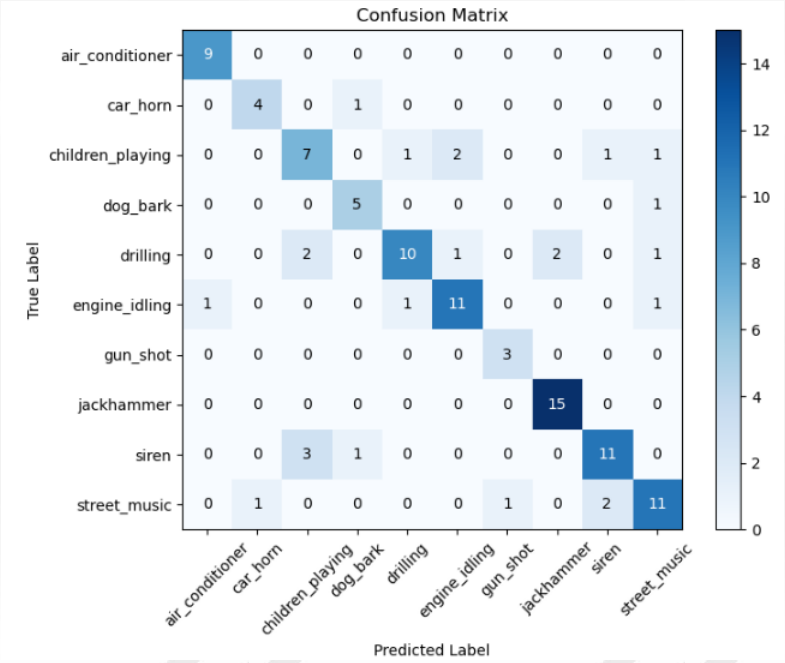
CNN: `Test Accuracy: 0.7954545454545454`

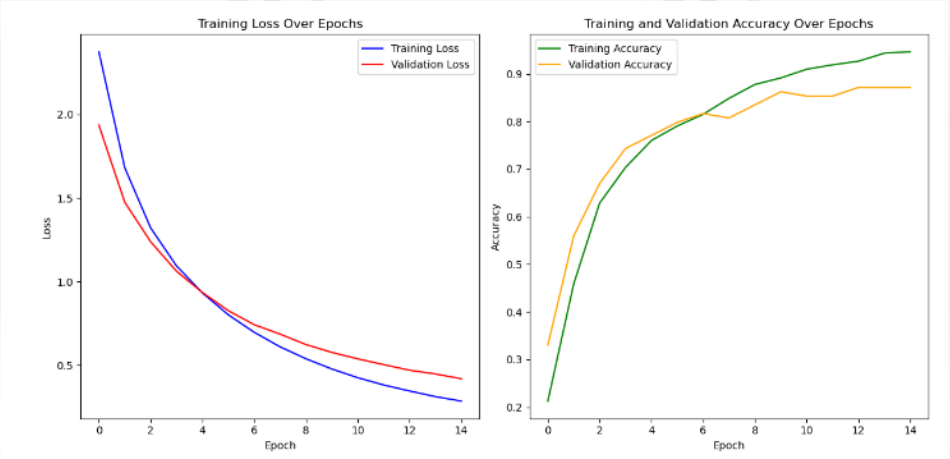




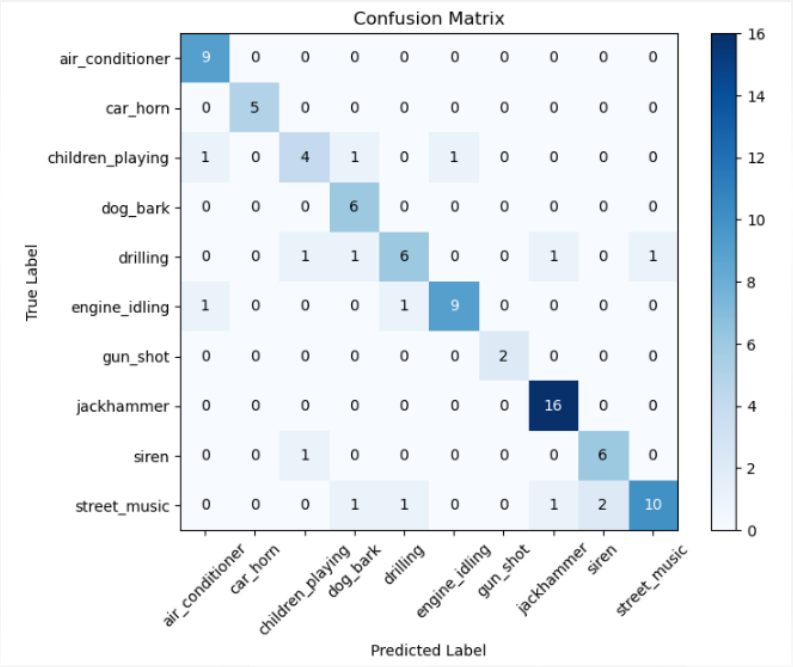
Resultados do Projeto:

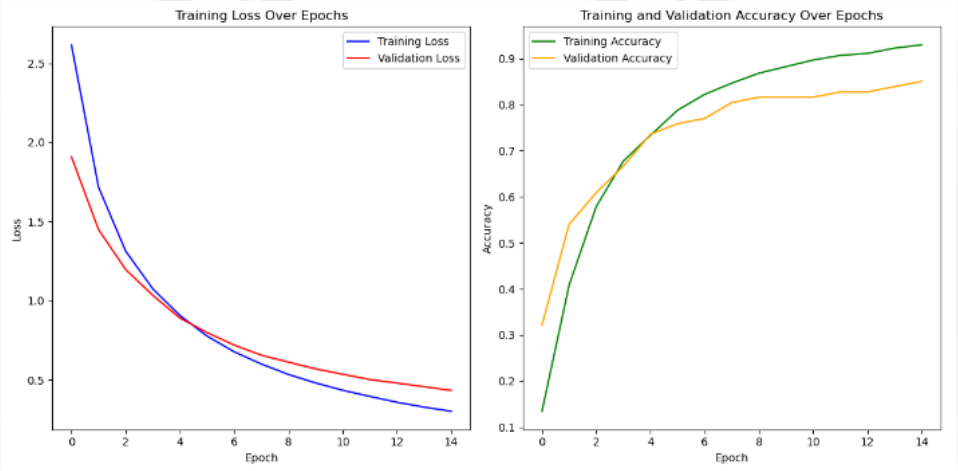
MLP com repetição e várias estatísticas: `Accuracy: 0.7818182110786438`



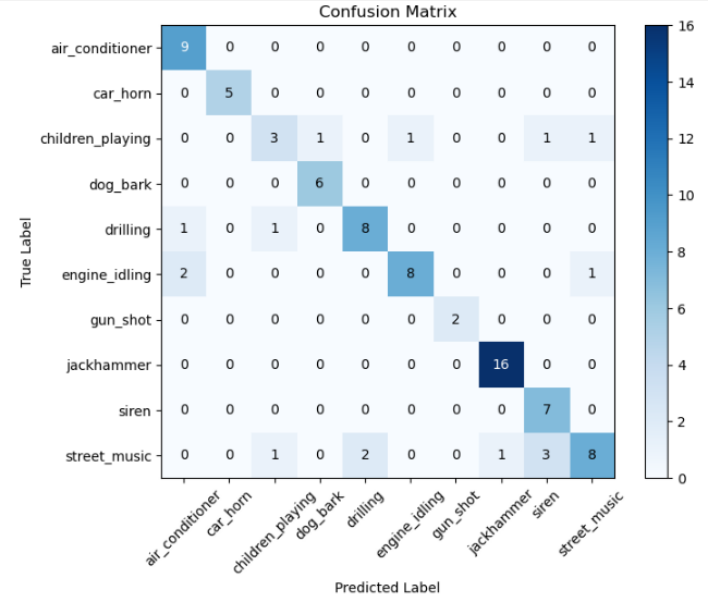


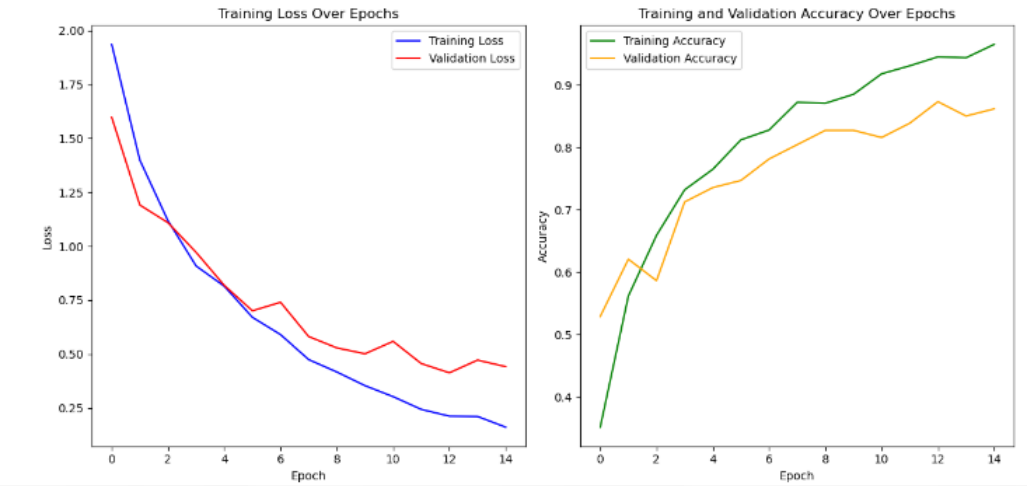
MLP com interpolação e várias estatísticas: `Accuracy: 0.8295454382896423`



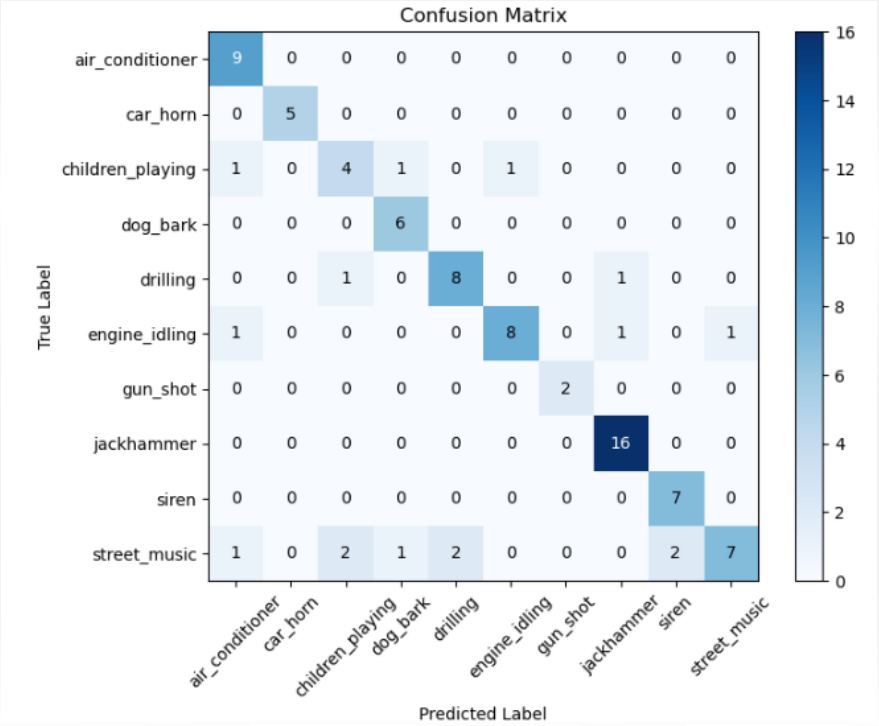


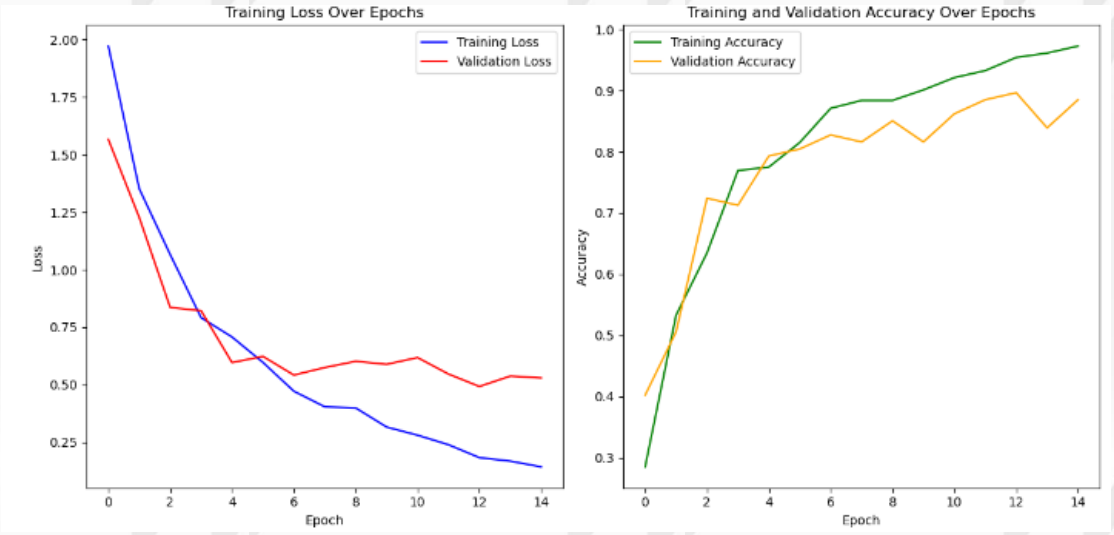
CNN com repetição: `Accuracy: 0.8181818181818182`



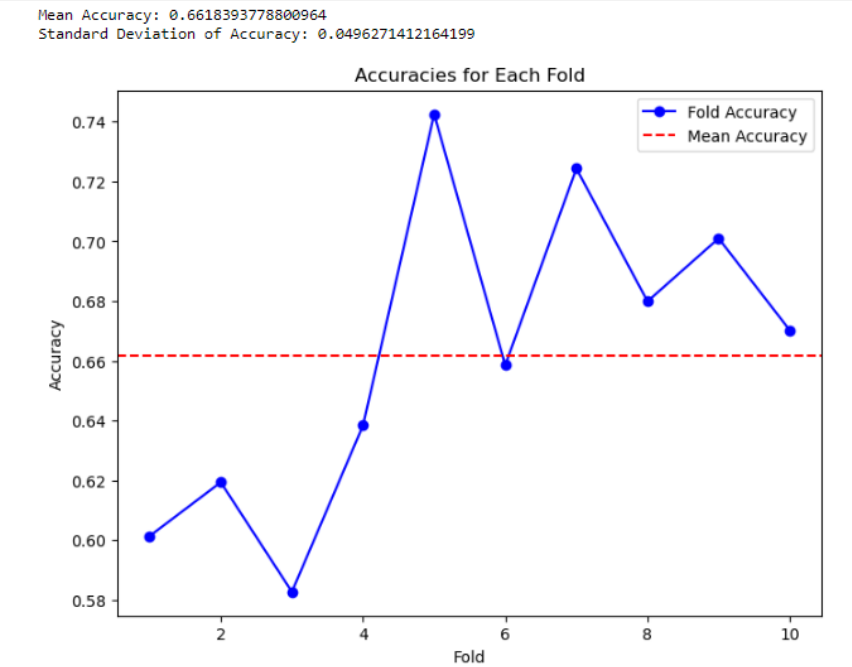


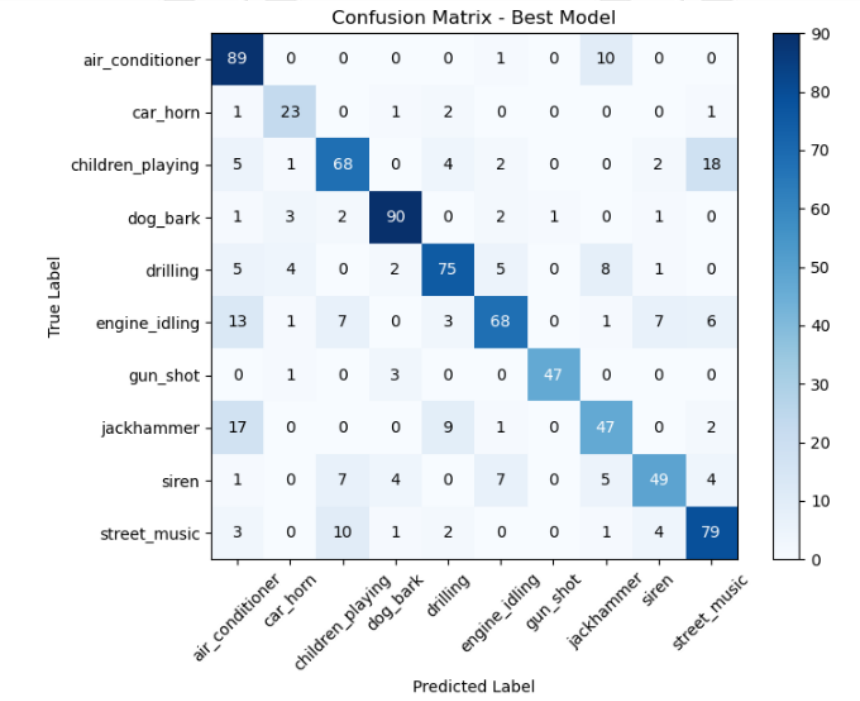
CNN com interpolação: `Accuracy: 0.8181818181818182`

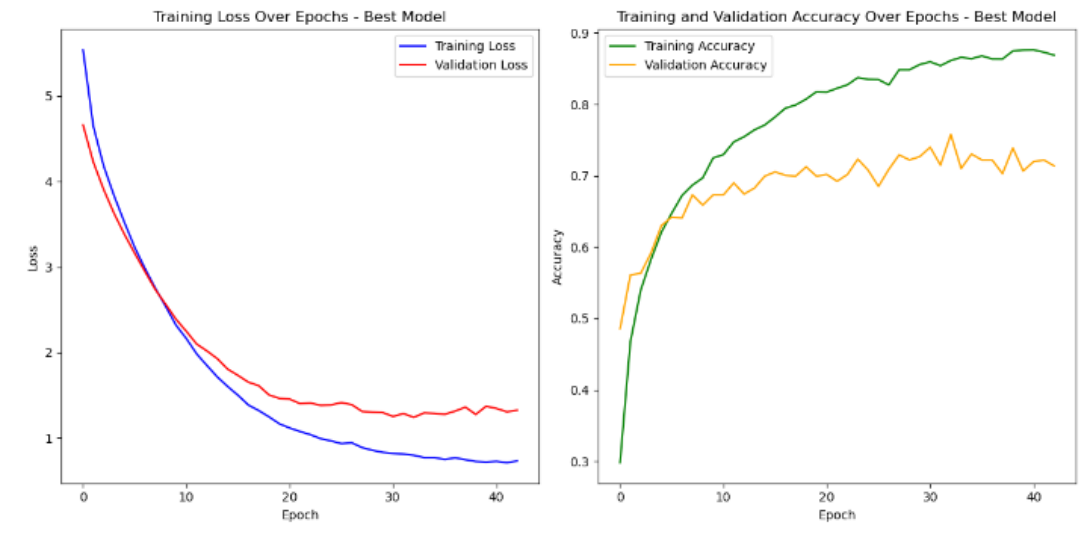




MLP aplicado ao dataset inteiro e com modelo mais complexo:



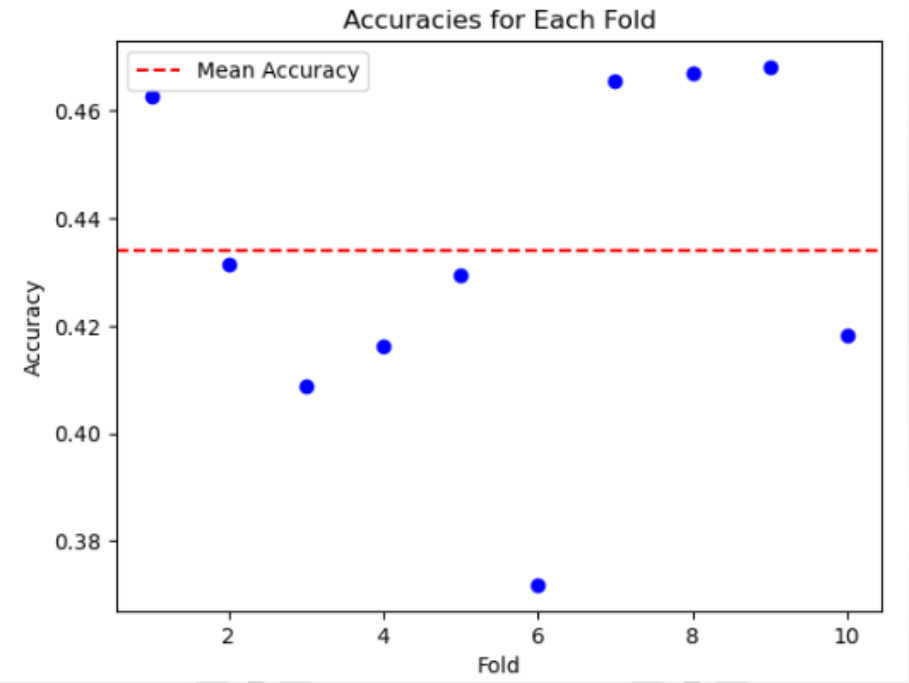




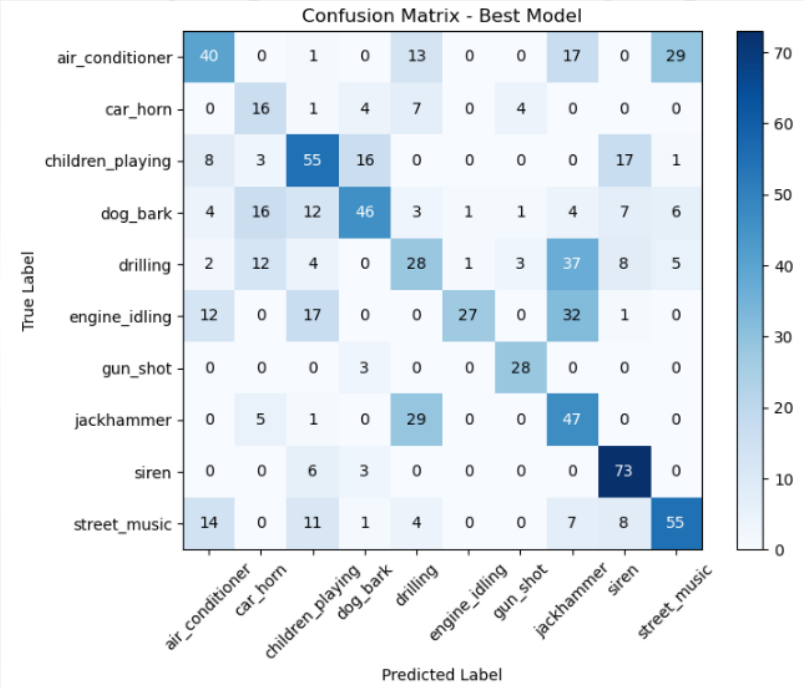
CNN aplicado ao dataset inteiro e com modelo mais complexo:

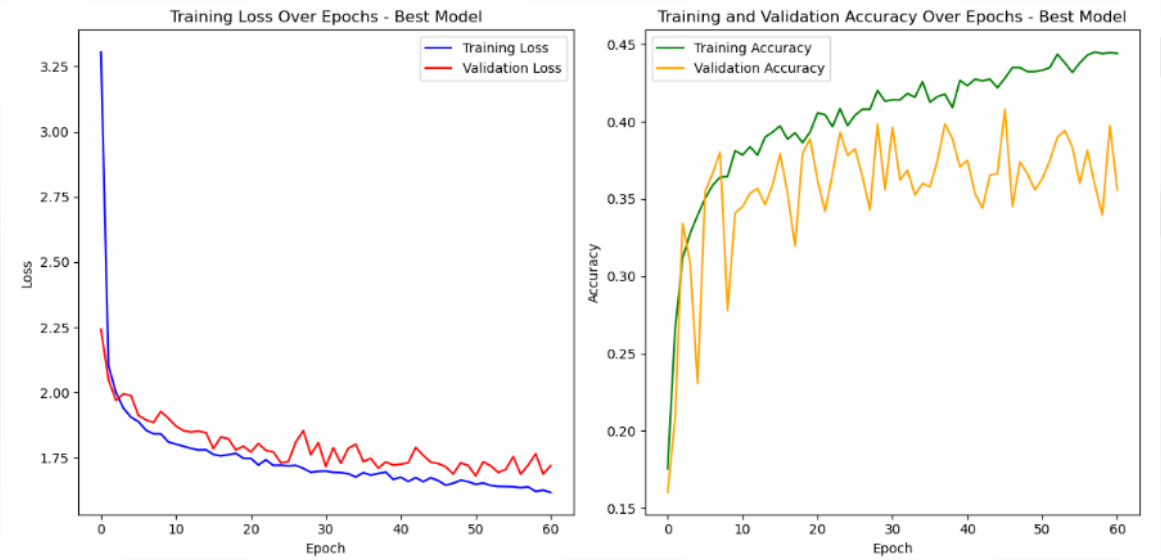
```
Mean Accuracy: 0.43387108743190766
Standard Deviation of Accuracy: 0.030271426948612963

<matplotlib.legend.Legend at 0x2b0324bbfd0>
```



```
Best Model Accuracy: 0.5085784196853638
```





5

Conclusões

Ao concluir a nossa exploração do problema de classificação em análise, é crucial refletir sobre o desempenho demonstrado pelos classificadores implementados e extrair insights significativos. O desenvolvimento deste projeto envolveu uma avaliação meticulosa de várias arquiteturas de aprendizado profundo, nomeadamente Perceptron Multicamada (MLP) e Rede Neural Convolucional (CNN), cada uma adaptada para abordar os desafios únicos apresentados pelo nosso conjunto de dados.

Os resultados obtidos neste projeto não se observaram nem extremamente desfavoráveis nem notavelmente excelentes.

Numa primeira fase, foi possível perceber que o uso de repetição e interpolação no pré-processamento de dados teve um impacto positivo, principalmente a interpolação que posteriormente foi escolhida como a estratégia adotada para o resto do projeto.

Ao examinarmos o desempenho do modelo MLP, é possível observar uma performance ligeiramente positiva. Esta conclusão é sustentada pela análise visual dos gráficos, nos quais se destaca a proximidade entre a accuracy de treino e validação, indicando a ausência de overfitting. Além disso, a discrepância entre as perdas de treino e validação, embora mais elevada na última, permanece próxima, sugerindo a ausência de underfitting. Essa convergência de valores reflete uma adaptação equilibrada do modelo aos dados, contribuindo para a sua estabilidade e capacidade de generalização. Por outro lado, o facto de apenas obtermos uma accuracy media de 66% quando foi aplicada cross validation mostra que o modelo nao tem uma boa capacidade de generalização.

Ao examinarmos o desempenho do modelo CNN, é possível observar uma performance moderada sem inclinação positiva para o lado negativo ou positivo. Esta conclusão é sustentada pela análise visual dos gráficos, nos quais se destaca a proximidade entre a accuracy de treino e validação, indicando a ausência de overfitting. Além disso, a discrepância entre as perdas de treino e validação, embora mais elevada na última, permanece próxima, sugerindo a ausência de underfitting. No entanto, conseguimos observar várias oscilações no gráfico de loss mostra que possivelmente o conjunto de dados de validação não é completamente representativa da variabilidade dos dados reais, uma vez que através do ajuste dos vários parâmetros não foi possível melhorar este facto sem que o modelo sofresse de overfit ou underfit. Para além disso, a presença de um valor de accuracy media de 43% ao longo do cross-validation mostra que o modelo nao apresenta uma boa generalização em relação aos dados.

Em conclusão, a análise abrangente dos modelos MLP e CNN revela um desempenho que se situa entre extremos, sem alcançar resultados excecionalmente desfavoráveis, mas também sem atingir níveis notáveis de excelência. Embora o modelo MLP tenha demonstrado uma adaptação equilibrada e estável aos dados, refletida na proximidade entre as métricas de treino e validação, a sua capacidade de generalização ficou aquém do desejado, evidenciada pela modesta média de precisão de 66% durante a validação cruzada. Por outro lado, o modelo CNN apresentou uma performance moderada, indicando a ausência de overfitting e underfitting, mas as oscilações nos gráficos de perda levantam questões sobre a representatividade do conjunto de dados de validação. A busca por uma configuração otimizada de parâmetros não conseguiu superar esse desafio sem comprometer a estabilidade do modelo. Em última análise, este projeto destaca a complexidade inerente à tarefa de classificação, apontando para futuras oportunidades de refinamento e investigação, seja na escolha de arquiteturas mais avançadas, no ajuste cuidadoso de hiperparámetros ou na exploração de estratégias de aumento de dados.