UNIversidade federal do Paraná

MATEUS FELIPE DE CÁSSIO FERREIRA

RELATÓRIO: Convolutional neural network



curitiba

2021mateus felipe de cássio ferreira

relatório: Convolutional neural network

Relatório apresentado como requisito parcial à conclusão da disciplina CI171 – Aprendizagem de Máquina, no Curso de Bacharelado em Informática Biomédica, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Orientador: Prof. Dr. Luiz Eduardo Soares de Oliveira

curitiba

2021

SUMÁRIO

[1 INTRODUÇÃO 3](#_Toc75730584)

[2 METODOLOGIA 4](#_Toc75730585)

[2.1 Lenet 5 4](#_Toc75730586)

[2.2 CNN IMPLEMENTADA 5](#_Toc75730587)

[*2.3 DATA AUGMENTATION* 7](#_Toc75730588)

[3 RESULTADOS 8](#_Toc75730589)

[3.1 LENET 5 8](#_Toc75730590)

[3.2 MATT CNN 10](#_Toc75730591)

[3.3 EXTRAÇÃO DE CARACTERÍSTICAS 12](#_Toc75730592)

[3.4 data augmentation 13](#_Toc75730593)

[4 CONCLUSÃO 16](#_Toc75730594)

# INTRODUÇÃO

Este relatório busca apresentar os resultados obtidos referentes ao terceiro Laboratório da disciplina de CI171 – Aprendizagem de Máquina, que consiste em implementar duas *Convolutional Neural Network* (CNN), funções de *Data Augmentation* e desenvolver o conceito de *Transfer Learning / Fine-Tuning*.

O objetivo deste relatório é o de analisar o desempenho de duas Redes Neurais Convolucionais a partir de diferentes parâmetros para a compilação e treinamento do modelo. Ainda, pretende-se realizar os mesmos testes para uma base de treinamento com e sem as funções de *Data Augmentation*.

Além disso, esse laboratório visa repetir os experimentos com duas redes pré-treinadas da ImageNet, que serão utilizadas como extratores de características para realizar a classificação da base de teste em um outro classificador.

# METODOLOGIA

Este relatório utilizou, como base de dados, imagens de meses do ano manuscritos em português. Uma vez que estamos trabalhando com imagens, uma rede neural do tipo CNN é uma metodologia usualmente utilizada para a classificação desse tipo de problema. Assim, foi utilizado 1.578 exemplos para treinamento do modelo e 401 exemplos para o teste.

No entanto, sabe-se que problemas que envolvam imagens necessitam de uma vasta quantidade de imagens na base de treinamento, visto que a CNN possui pesos que devem ser ajustados durante a fase de treinamento do modelo. Nesse sentido, uma das técnicas utilizadas para contornar esse problema é o chamado *Data Augmentation*, uma técnica de regularização que consiste em aumentar os exemplos de uma base de treinamento ao adicionar ruídos, rotações, translações e mudanças de escala, por exemplo. Essa técnica permite, assim, a redução de erros de generalização.

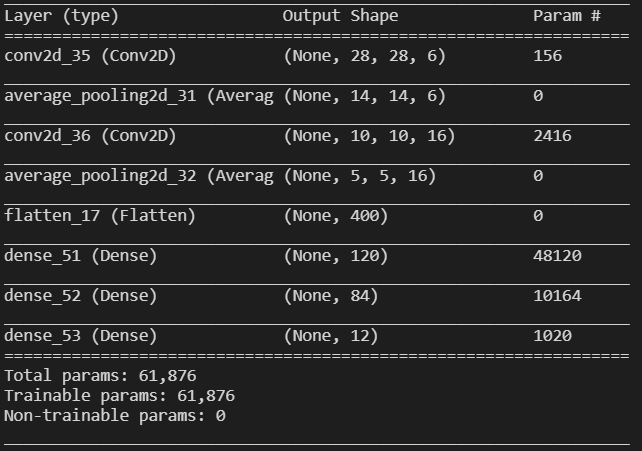
## Lenet 5

A LeNet foi uma CNN proposta no ano de 1998 por Yann LeCun, Leon Bottou, Yoshua Bengio, e Patrick Haffner. A estrutura dessa CNN conta com três camadas convolucionais, duas camas de agregação e duas camadas totalmente conectadas, totalizando sete camadas ao total. A Figura 1 apresenta a forma como a LeNet 5 foi implementada para o problema proposto. O parâmetro *input\_shape* contém uma lista com os valores (32, 32, 1). Além disso, na última camada, o parâmetro do número de classes (*num\_classes*) é igual a doze (doze meses do ano).

A Figura 2, por outro lado, apresenta a arquitetura da LeNet 5 que foi implementada utilizando a biblioteca “Keras” para redes neurais.

 FIGURA 1 – IMPLEMENTAÇÃO DA LENET 5

FONTE: O autor (2021).

FIGURA 2 – ARQUITETURA DA LENET 5

FONTE: O autor (2021).

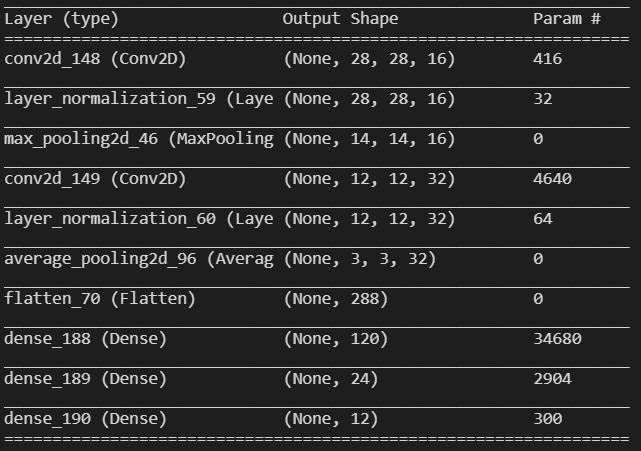
## CNN IMPLEMENTADA

A estrutura da CNN implementada, a MATT CNN, conta com três camadas convolucionais, duas camas de agregação, duas camadas totalmente conectadas, e duas camadas de normalização, totalizando nove camadas ao total. A Figura 3 apresenta a forma como a MATT CNN foi implementada para o problema proposto. O parâmetro *input\_shape* contém uma lista com os valores (32, 32, 1).

A Figura 4, por outro lado, apresenta a arquitetura CNN implementada.

 FIGURA 3 – IMPLEMENTAÇÃO DA MATT CNN

FONTE: O autor (2021).

FIGURA 4 – ARQUITETURA DA MATT CNN

FONTE: O autor (2021).

## *DATA AUGMENTATION*

A Figura 5 apresenta uma parte da função que foi construída para realizar o processo de *Data Augmentation*. Nessa função foi feito um *“flip”* vertical e horizontal e três rotações (nos ângulos de 5, 10 e 15) para cada imagem do conjunto de treinamento.

FIGURA 5 – IMPLEMENTAÇÃO DA FUNÇÃO DE DATA AUGMENTATION

FONTE: O autor (2021).

# RESULTADOS

## LENET 5

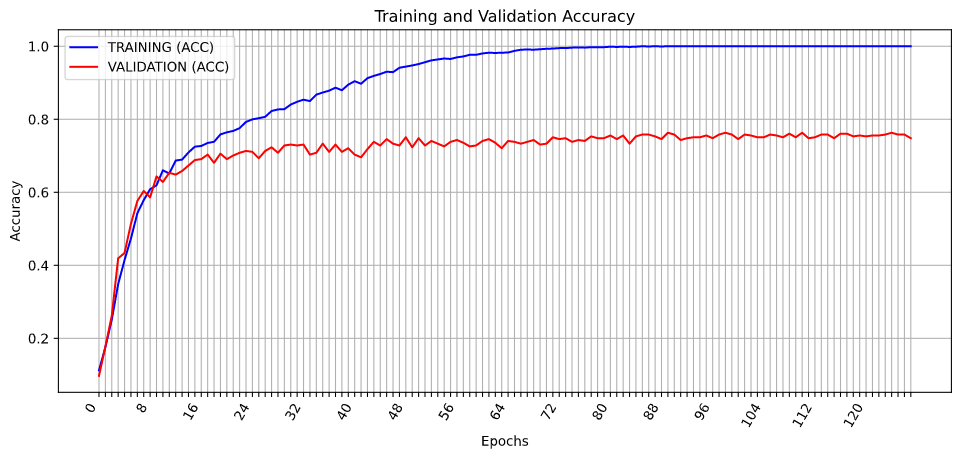
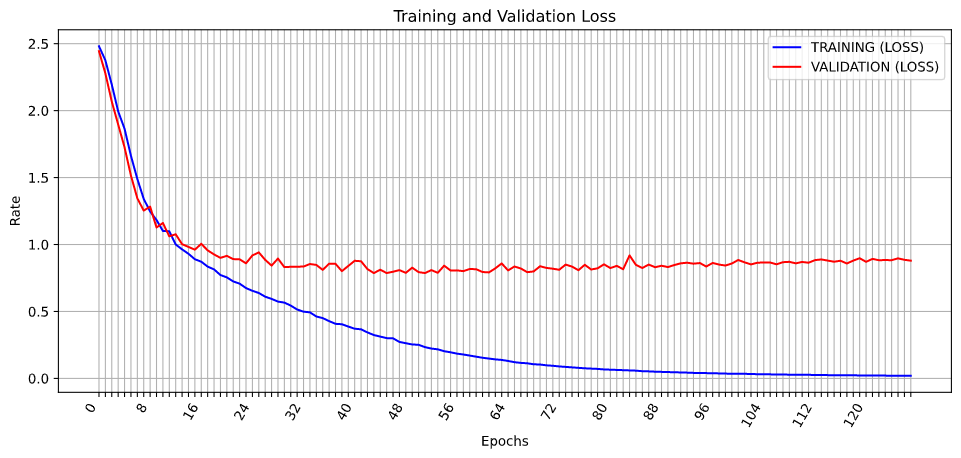
A Tabela 1 apresenta os resultados obtidos para diferentes situações em que foram alterados os parâmetros *batch\_size, n\_epochs e learning\_rate*. Nota-se que o melhor resultado, considerando apenas a medida da acurácia, foi a SIT09 (ou SIT10). No entanto, a Figura 6 apresenta as medidas de acurácia e os erros cometidos pelo modelo durante o treino e validação. Nota-se que o valor do erro cometido pelo modelo durante a validação é muito maior do que o erro do modelo durante a fase de treinamento. Essa situação indica que o modelo entrou em *overfitting* e, apesar de ter o melhor desempenho no valor da acurácia, na prática, o modelo não consegue generalizar tão bem as entradas quando é preciso fazer a validação.

Nesse sentido, embora a situação SIT01 não tenha o melhor valor de acurácia, esse é o modelo em que o classificador consegue generalizar melhor a situação de validação do modelo. A Figura 7 apresenta os resultados obtidos da SIT01. Nota-se que os valores de erro da fase de treinamento e validação estão decaindo e convergindo quase que juntas. Essa é uma situação, aliado ao fato de que os valores de acurácia para os dois casos também estão convergindo para valores muito próximos, demonstra que o modelo consegue generalizar bem a base de validação e é um bom candidato para a classificação na prática.

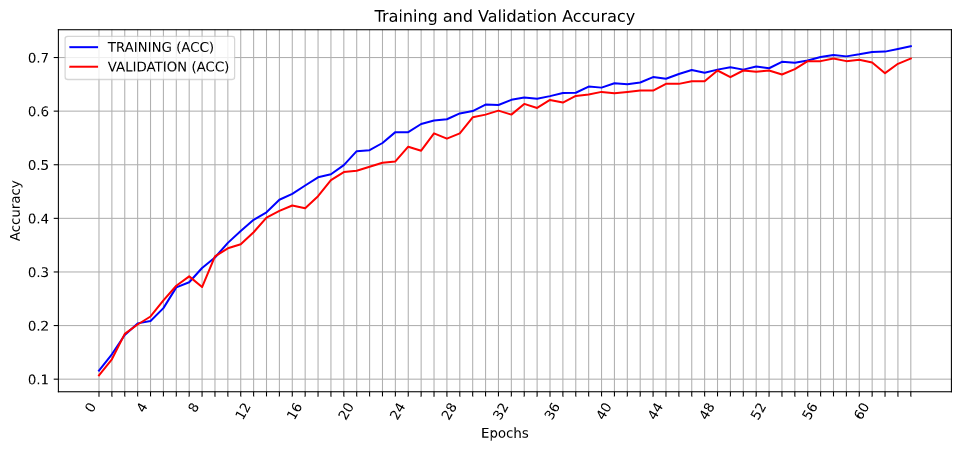
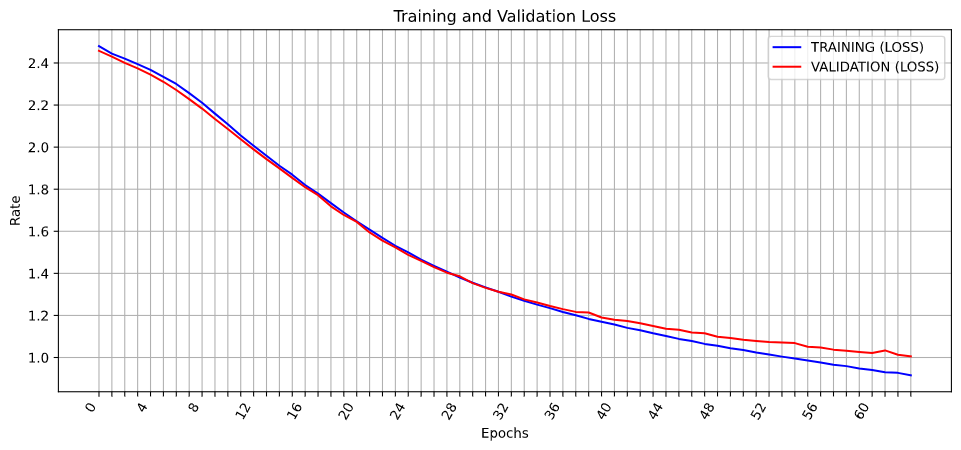
TABELA 1 – RESULTADOS OBTIDOS PARA DIFERENTES SITUAÇÕES

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SITUAÇÕES** | **BATCH SIZE** | **N EPOCHS** | **LEARNING RATE** | **TEST LOSS** | **TEST ACCURACY** |
| SIT01 | 64 | 64 | 0,01 | 0,968 | 0,693 |
| SIT02 | 128 | 64 | 0,01 | 1,275 | 0,613 |
| SIT03 | 128 | 128 | 0,01 | 1,011 | 0,673 |
| SIT04 | 128 | 128 | 0,001 | 2,326 | 0,251 |
| SIT05 | 64 | 64 | 0,001 | 2,403 | 0,211 |
| SIT06 | 64 | 64 | 0,1 | 0,951 | 0,768 |
| SIT07 | 128 | 64 | 0,1 | 0,818 | 0,741 |
| SIT08 | 64 | 128 | 0,1 | 0,974 | 0,775 |
| SIT09 | 128 | 128 | 0,1 | 0,935 | 0,758 |
| SIT010 | 64 | 256 | 0,1 | 1,081 | 0,758 |

FONTE: O autor (2021).

FIGURA 6 – RESULTADOS OBTIDOS PARA A SIT09

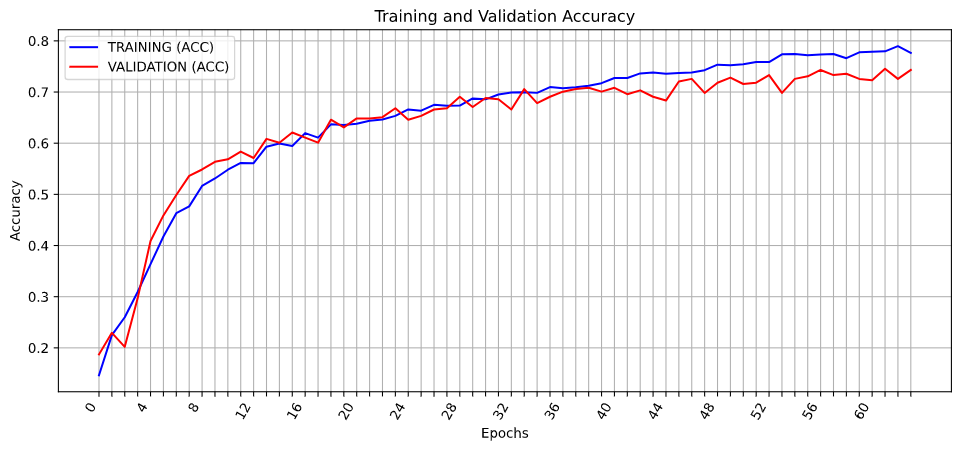
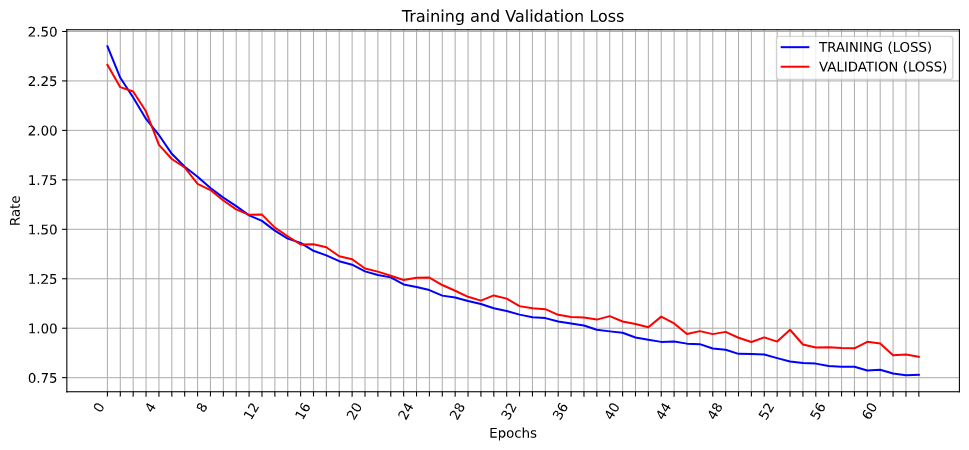
FONTE: O autor (2021).

FIGURA 7 – RESULTADOS OBTIDOS PARA A SIT01

FONTE: O autor (2021).

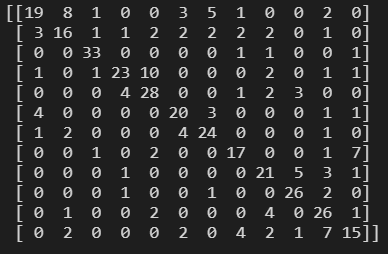
## MATT CNN

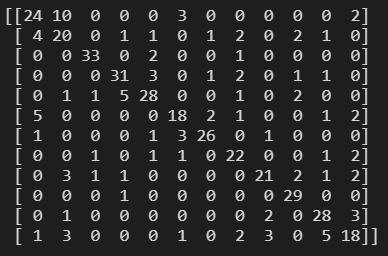
Os resultados para a MATT CNN estão apresentados na Figura 8. O valor máximo de acurácia atingido por esse modelo foi de 0,743, com um valor de *“test loss”* igual a 0,855. Esse valor foi obtido mudando os parâmetros de função de ativação, adicionando e removendo determinadas camas, mudando o tamanho do kernel e para diferentes valores para o parâmetro *optimizer* na compilação da CNN. Apesar de ser um valor pouco superior ao LeNet 5, tentei fazer com que esse modelo apresentasse uma convergência semelhante ao modelo base. Outras medidas de acurácia acima de 0,8 foram obtidas, mas ao verificar o gráfico para comparação entre as acurácias de treinamento e validação, bem como os erros nessas duas fases, o modelo entrou em *overfitting*.

FIGURA 8 – RESULTADOS OBTIDOS PARA A MATT CNN

FONTE: O autor (2021).

A Figura 9 apresenta a comparação entre as duas matrizes de confusão para a LeNet5 e a MATT CNN.



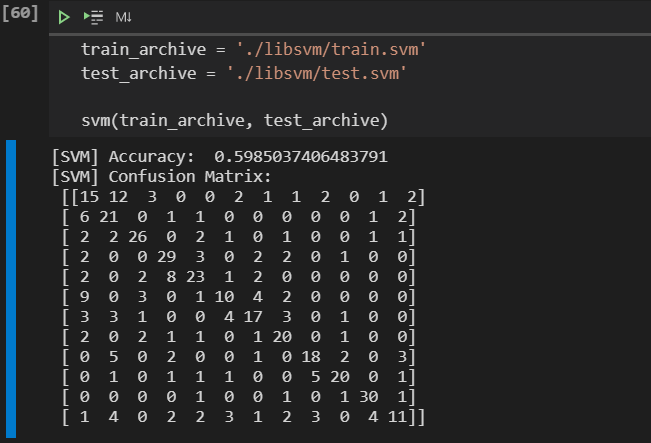
FIGURA 9 – RESULTADOS OBTIDOS

a) LeNet5 b) LeNet5

FONTE: O autor (2021).

## EXTRAÇÃO DE CARACTERÍSTICAS

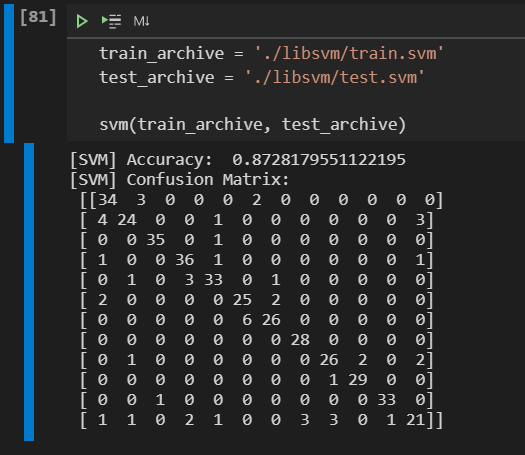
Uma das redes pré-treinadas escolhidas para reaplicar os testes foi a rede InceptionV3. Essa rede foi utilizada para extrair as características dos arquivos de treino e teste para utilizarmos o arquivo de saída, no formato LIBSVM, em um outro classificador. Nesse caso, o classificador escolhido foi um *Support Vector Machine*. A Figura 10 apresenta o resultado obtido de acurácia e a matriz de confusão para esse classificador.

FIGURA 10 – RESULTADOS OBTIDOS COM A INCEPTION V3 PARA EXTRAÇÃO DE CARACTERÍSTICAS E CLASSIFICAÇÃO EM UM SVM

FONTE: O autor (2021).

Por outro lado, foi utilizado uma segunda rede pré-treinada para reaplicar os testes, chamada de VGG16. Essa rede também foi utilizada para extrair as características dos arquivos de treino e teste para utilizarmos o arquivo de saída, no formato LIBSVM, no classificador SVM. A Figura 11 apresenta o resultado obtido de acurácia e a matriz de confusão para esse classificador. Nota-se que a medida de acurácia para esse classificador, utilizando a rede pré-treinada VGG16 foi muito superior quando comparado o desempenho do mesmo classificador utilizando a rede InceptionV3.

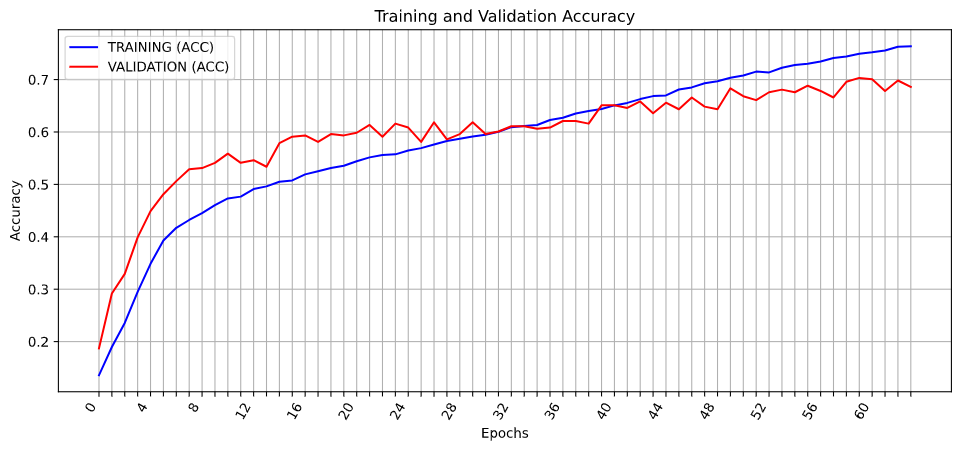
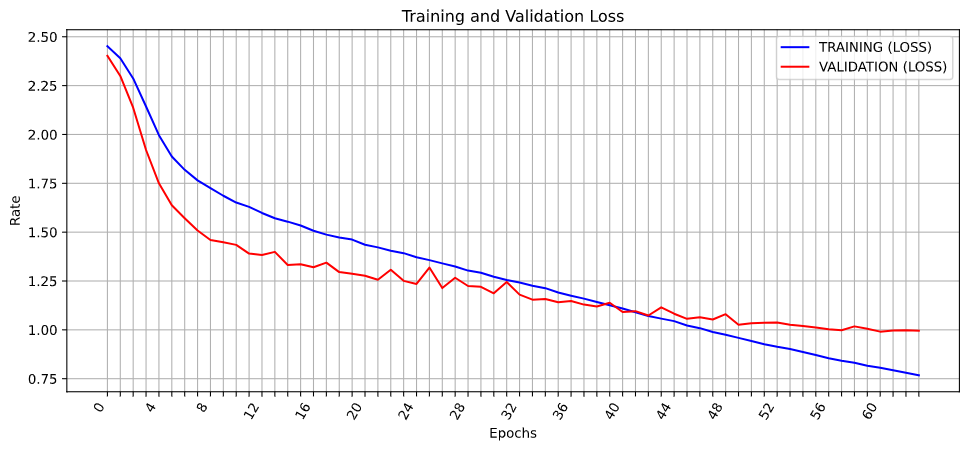
FIGURA 11 – RESULTADOS OBTIDOS COM A VGG16 PARA EXTRAÇÃO DE CARACTERÍSTICAS E CLASSIFICAÇÃO EM UM SVM



FONTE: O autor (2021).

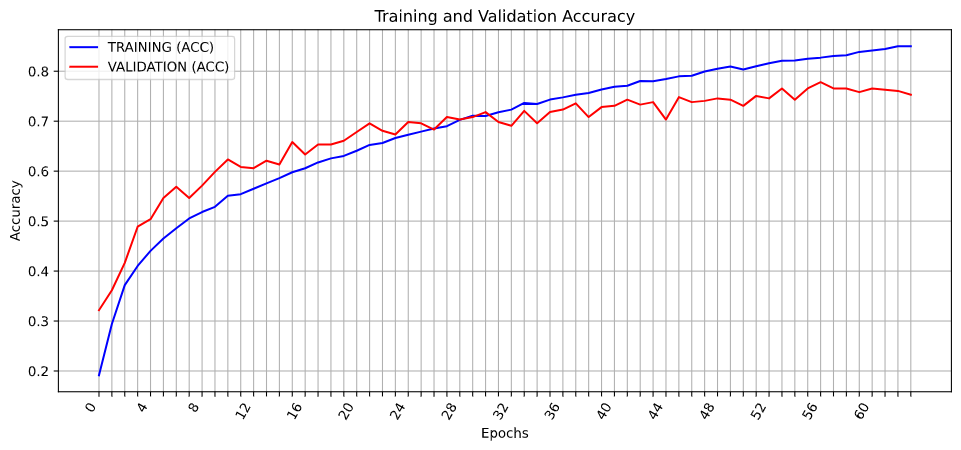
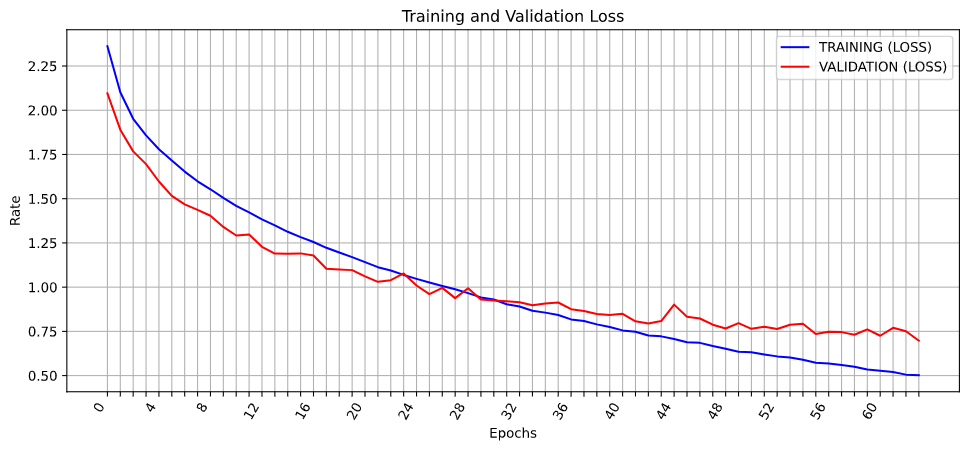
## data augmentation

A Figura 12 apresenta os resultados obtidos utilizando a função de Data Augmentation para o LeNet 5.

FIGURA 12 – RESULTADOS OBTIDOS COM DATA AUGEMNTATION PARA A LENET5

FONTE: O autor (2021).

A Figura 13 apresenta os resultados obtidos utilizando a função de Data Augmentation para o MATT CNN.

FIGURA 13 – RESULTADOS OBTIDOS COM DATA AUGEMNTATION PARA A LENET

FONTE: O autor (2021).

# CONCLUSÃO

A partir dos resultados do experimento nota-se que existe um significativo impacto na escolha dos parâmetros e configurações utilizadas (camadas) em uma CNN. De maneira geral, as CNN apresentaram bons resultados quando comparado com alguns outros classificadores. Foi possível notar também que o uso de uma CNN para extração de vetores de características pode ser uma solução muito benéfica para alguns problemas, visto que nesse laboratório foi possível obter uma acurácia acima de 0,8 para o problema utilizando o classificador SVM a partir da extração de um vetor de características pelo VGG61.

De maneira geral, a CNN obtém ótimos resultados e que se tornam ainda mais positivos ao utilizarmos as funções de Data Augmentation para a resolução do problema.