

# **Reconhecimento de atividade humana usando deep learning e video datasets**

Projeto Prático de Inteligência Artificial

2021

|  |  |
| --- | --- |
| **Autores:** | Josué Braz nº 35558  Diogo Araújo nº 35128 |
| **Docente:** | Prof. José Manuel Torres |

**Índice**

[Reconhecimento de atividade humana usando deep learning e video datasets 1](#_Toc20680)

[1. Introdução 3](#_Toc16448)

[2. Descrição do Problema 4](#_Toc31824)

[3. Estado da Arte 5](#_Toc31655)

[4. Descrição do Trabalho 6](#_Toc10810)

[5. Análise de Resultados 10](#_Toc811)

[6. Conclusões e Perspetivas de Desenvolvimento 13](#_Toc19214)

[7. Referências 14](#_Toc14642)

# Introdução

Este documento dedica-se à descrição detalhada da elaboração do projeto final para a disciplina de Inteligência Artificial, que consiste em explorar esquemas de Deep- Learning, manipular datasets em linguagem Python, com auxílio do Keras e de outras bibliotecas. Através de vários datasets foi-nos possível treinar vários modelos cujo foco é o reconhecimento de atividade humana

## Motivação

O projeto desenvolvido aprofunda o estudo de Machine Learning, lecionado ao longo das aulas de Inteligência Artificial do Mestrado em Engenharia Informática, proporcionando-nos um melhor conhecimento para o desenvolvimento deste projeto e a compreensão necessária para alguns dos esquemas trabalhados.

Este projeto trata de um conjunto de dados sobre atividade humana, dados estes interessantes para serem usados em reconhecer e prever ações através de imagens, que potencialmente possui vários usos.

## Objetivos

Ao escolher este conjunto de dados, pretendíamos, com o auxílio de ferramentas e esquemas de Deep-Learning, como a biblioteca Keras, tratar de avaliar a predição de atividades humanas através de um vídeo, verificando isto através dos diversos valores apresentados para cada frame do mesmo, a sua comparação e como estes podem ter impacto na decisão final. Após isto, é-nos possível obter uma percentagem aproximada da veracidade dos dados através do seu nível de precisão e do seu nível de perda. Após esta avaliação e tratamento rígido dos dados, é possível verificar a probabilidade de a previsão estar certa.

# Descrição do Problema

O desafio principal deste projeto envolve a análise e comparação de vários métodos de classificação de vídeo, que fazem uma previsão da atividade com base em modelos já previamente treinados. Para este trabalho foram implementados, estudados e comparados 2 métodos, o método Single-Frame CNN e CNN com LSTMs.

O conjunto de dados escolhido foi um grupo de imagens que possuem variadas atividades humanas, com 50 categorias de ações. A maioria dos data sets não são realistas e são encenados por atores.

Posteriormente, em código Python, foi necessário carregar estes dados, para então os visualizar graficamente e manipular. Para isto, foram utilizadas várias bibliotecas, referidas mais abaixo, que proporcionaram a visualização dos dados em diferentes tipos de gráficos para melhor compreender o problema. Após a compreensão dos dados deste conjunto, procedeu-se ao real objetivo, que é avaliar a precisão destes dados. Após importar os dados é necessário, então, proceder à implementação do modelo de redes neuronais em Keras (modelo NN).

Com o modelo pronto a ser usado é necessário de seguida compila-lo. Esta compilação utiliza o Tensorflow que oferece várias bibliotecas para executar esta função. Após a sua compilação, o modelo sequencial está pronto a ser trabalhado com os dados. Nesta fase podemos então calcular as percentagens de precisão do modelo, ou seja a fiabilidade dos dados e o valor das perdas. A precisão varia á medida que trabalhamos e ajustamos o modelo ao conjunto de dados. Os valores são posteriormente representados graficamente o que nos permite fazer uma avaliação dos mesmos.

# Estado da Arte

Sabemos que a inteligência artificial pode ser aplicada em diversas áreas e contextos e que diversos estudos já foram realizados, e sendo o nosso tema relacionado com reconhecimento de imagem, esperava-mos que existisse muita informação. Como mencionado previamente neste relatório apenas serão analisados 2 métodos.

O single frame CNN foi o primeiro método estudado e analisado, ele consiste com base em algumas frames espalhadas pelo vídeo em análise, calcular a probabilidade de uma determinada atividade estar a ocorrer, fazendo posteriormente a análise a média de todas as probabilidades individuais, obtendo assim um vetor de probabilidades. No geral é um método que possui um bom desempenho e resultados promissores. Neste trabalho o CNN foi implementado com um dataset que possui vários vídeos demonstrando atividades humanas, nomeadamente as usadas: corridas de cavalos, andar de baloiço, passear um cão, tai chi.

No caso da arquitetura CNN com LSTMs, é feito o uso de camadas CNN para extrair recursos em dados de entrada combinados com LSTMs para oferecer suporte á previsão de sequência. Isto difere do CNN pois ele não é recorrente, ou seja, não retém memória de padrões de séries temporais anteriores. Ele só consegue treinar com base em dados inseridos num determinado instante. Em suma, a arquitetura CNN com LSTMs foi desenvolvida para problemas de previsão visual de séries temporais com o objetivo de gerar descrições provenientes das sequências de imagens.

# Descrição do Trabalho

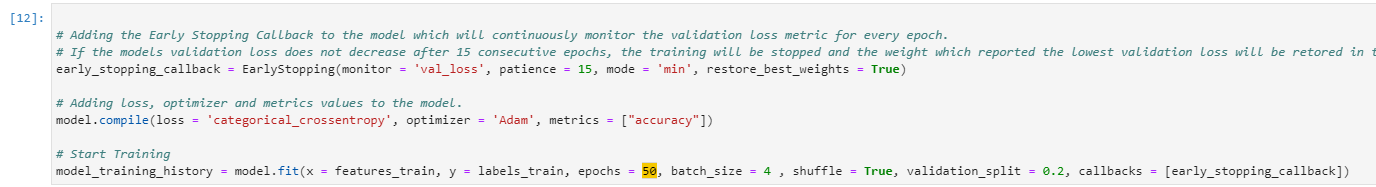
Como início foi implementado o algoritmo de de classificação de vídeo single-frame CNN, desenvolvido em python através da biblioteca keras. De maneira a organizar o código e ter uma abordagem por blocos foi decidido que este ia ser desenvolvido via Jupiter. Como foi mencionado previamente foi necessário fazer uso de um dataset de vídeos de maneira a testar os vários métodos de reconhecimento de atividade humana. Para o tal, foi utilizado o Human activity Recognition using smatphones data set, pelo UCI Machine Learning repository. Posteriormente foi carregado este dataset para o Jupiter.

Após ter sido implementado o código foram selecionadas 4 classes: WalkingWithdog, TaiChi, Swing e HorseRace. Estas são as classes para quais o modelo será treinado, usando vídeos relacionados com o tópico de cada uma.



*Figura 1- modelos a ser treinados*

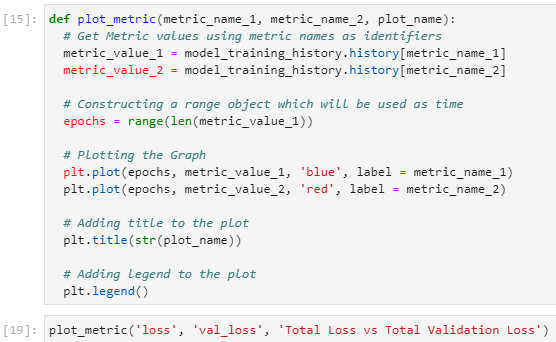
Após extrair as classes foi necessário treinar o modelo. Para o treino dos modelos, foi necessário definir um número de epochs, ou épocas de treino. Quantas mais épocas escolhermos mais accurate o modelo vai ser. Para este projeto foram selecionadas 50 épocas pois aparentam ser suficientes para obter resultados satisfatórios e não levam muito tempo a serem processadas.



*Figura 2- treinar os modelos*

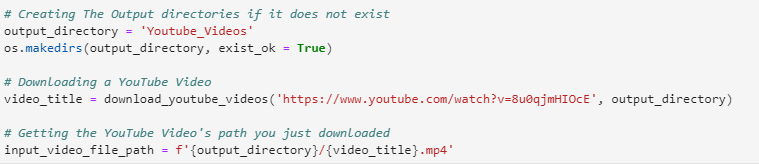
Após o treinos dos nossos modelos o algoritmo procede a traçar os gráficos de resultados dos mesmos.

As métricas que tivemos em conta foram a accuracy e as perdas.



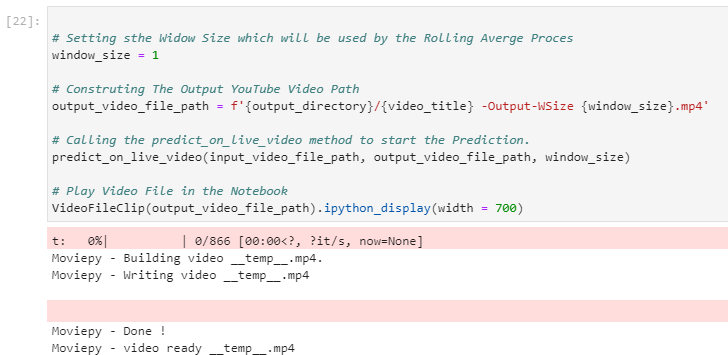
*Figura 3- traçar os gráficos*

Após termos traçado as métricas do algoritmo para gráficos temos então os modelos prontos para fazer previsões em tempo real com vídeos de teste. Para isto foi feito o download de vídeos da plataforma Youtube.



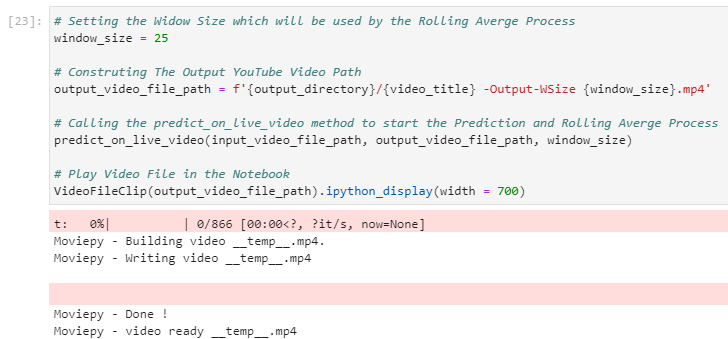
*Figura 4- download vídeo youtube*

Posteriormente, após ter o video guardado localmente e preparado para ter as labels colocadas, prosseguiu-se á previsão do mesmo utilizando a Moving average e sem Moving average. Se a window\_size for colocada com o valor 1, a função comporta-se como um classificador normal para prever frames de vídeos.



*Figura 5- previsão sem moving average*

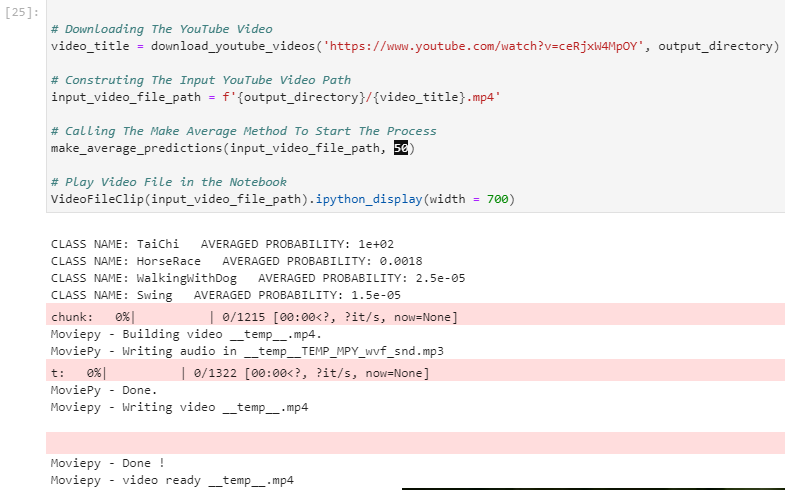
Utilizando Moving Average:



*Figura 6- previsão com moving average*

Apesar de os resultados não serem perfeitos pode-se observar que são bastante melhores que o método que não usa moving average.

De seguida foi utilizado então o método single-frame CNN. A função dá um output que faz uma unica previsão para o vídeo todo, ou seja, pega em N frames do vídeo inteiro e faz previsões. Posteriormente o algoritmo faz a média das previsões dessas N frames e dá-nos a previsão final da atividade desse vídeo.



*Figura 7- previsão CNN*

**CNN com LSTMs**

Após desenvolvermos o single frame CNN procedeu-se á implementação de um CNN LSTM network model, esta arquitetura envolve usar camadas CNN para a extração de dados combinados com LSTMs para dar suporte á previsão de sequência.

CNN LSTMs foram desenvolvidos para problemas de previsão de séries temporais e a aplicação de geração de descrições textuais a partir de sequência de imagens.

O modelo CNN LSTM lerá subsequências da sequência principal em blocos, extrairá recursos de cada bloco e, em seguida, permitirá que o LSTM interprete os recursos extraídos de cada bloco.

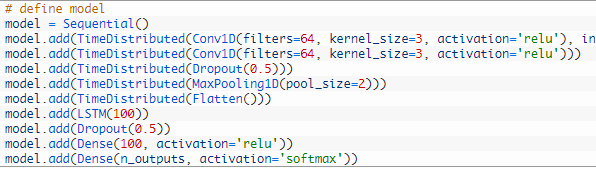
Uma abordagem para implementar este modelo é dividir cada janela de 128 etapas de tempo em subsequências para o modelo CNN processar. Por exemplo, as 128 etapas de tempo em cada janela podem ser divididas em quatro subsequências de 32 etapas de tempo.



*Figura 8-data subsequencias*

Podemos então definir um modelo CNN que espera ler em sequências com um comprimento de 32 intervalos de tempo e nove recursos.

Todo o modelo CNN pode ser empacotado numa camada TimeDistributed para permitir que o mesmo modelo CNN leia em cada uma das quatro subsequências na janela. Os recursos extraídos são então nivelados e fornecidos ao modelo LSTM para leitura, extraindo os seus próprios recursos antes que um mapeamento final para uma atividade seja feito.



*Figura 9- definir modelo*

Uma extensão adicional da ideia do CNN LSTM é realizar as convoluções do CNN como parte do LSTM. Essa combinação é chamada de LSTM convolucional ou, abreviadamente, ConvLSTM e, como o CNN LSTM, também é usado para dados espaço-temporais.

Ao contrário de um LSTM que lê os dados diretamente para calcular o estado interno e as transições de estado, e ao contrário do CNN LSTM que está interpreta a saída dos modelos CNN, o ConvLSTM usa convoluções diretamente como parte da leitura de entrada nas próprias unidades LSTM

A biblioteca Keras fornece a classe ConvLSTM2D que suporta o modelo ConvLSTM para dados 2D. Ele pode ser configurado para classificação de série temporal multivariada 1D.



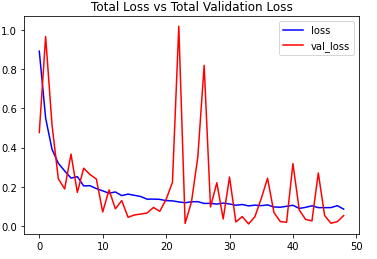
*Figura 10- definir modelo conv2D*

Podemos então avaliar os modelos como fizemos anteriormente.

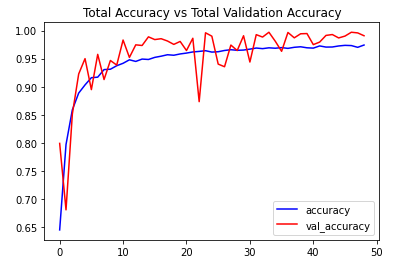
# Análise de Resultados

Começando pelo single-frame CNN, utilizando as classes referidas anteriormente podemos observar os seguintes resultados.

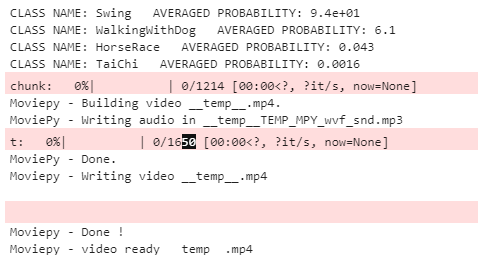
**Épocas = 50**



*Figura 11 - Modelo de Perdas*



*Figura 12- Modelo de Precisão*



*Figura 13- resultados CNN*

Como pode ser observado, ao utilizar um vídeo do Youtube de um baloiço como input, o algoritmo CNN, classifica com 94% de certeza, que se trata de facto de um baloiço representado no vídeo. Também é possível observar que ele deteta uma probabilidade de 6% de o vídeo se tratar de passear um cão, mas como a probabilidade é muito inferior á anterior, é descartada. As outras atividades possuem um nivel de certidão semelhante que ronda os 95-96%, á exceção do Tai-chi que surpreendentemente atingiu os 100%.

# Conclusões e Perspetivas de Desenvolvimento

# Referências

1. M. Thirugnanam et al., "Improving the Prediction Rate of Diabetes Diagnosis Using Fuzzy, NeuralNetwork, Case Based (FNC) Approach."Procedia Engineering, Vol.38, (2012). pp. 1709-118,.
2. H. R. Marateb et al., "A hybrid intelligent system for diagnosing microalbuminuria in type 2",(2014). pp. 34-42,.
3. J. A. Torkestani and G. P. Elham, "A learning automata-based blood glucose regulation mechanismin type 2 diabetes", Control Engineering Practice, Vol. 26, (2014). pp. 151-159.
4. S. Abu Naser, I. Zaqout, M. A. Ghosh, R. Atallah and E. Alajrami, “Predicting Student Performance Using Artificial Neural Network: in the Faculty of Engineering and Information Technology”, International Journal of Hybrid Information Technology, vol. 8, no. 2, (2015), pp. 221-228.
5. El\_Jerjawi, Nesreen & Abu-Naser, Samy. (2018). Diabetes Prediction Using Artificial Neural Network. Journal of Advanced Science. 124. 1-10.