**Aprendizado de Máquina 2**

**Trabalho 2**

Classificação com CNN

**Professor:** Prof. Dr. Murilo Coelho Naldi

Gabriel Lourenço de Paula Graton - 800432

Vitor Matheus da Silva - 800260

Rafael Naoki Arakaki Uyeta - 800207

São Carlos

2 de fevereiro de 2024

**Índice**

[**1. Introdução 3**](#_heading=h.gjdgxs)

[**2. Escolha do Conjunto de Dados 3**](#_heading=h.30j0zll)

[**3. Estudo do Conjunto de Dados 4**](#_heading=h.1fob9te)

[3.1. Descrição do conjunto de dados 4](#_heading=h.3znysh7)

[3.2. Atributos 4](#_heading=h.2et92p0)

[**4. Tratamento de Dados 8**](#_heading=h.nwd5lnn4vccz)

[**5. CNN. 11**](#_heading=h.d95jmshk2hma)

[**6.Resultados 18**](#_heading=h.ubr3m3669u5f)

[**7. Conclusão 20**](#_heading=h.1t3h5sf)

[**9. Bibliografia 21**](#_heading=h.h4cfzddot9uo)

## **1. Introdução**

Vídeo: <https://drive.google.com/file/d/1sHqyJIaQb_uOe4UjNGk7teKewZmf_uqG/view?usp=drive_link>

Esse trabalho tem como objetivo exercitar os conceitos aprendidos na disciplina de Aprendizado de Máquina 2 referente aos tópicos de redes neurais, mais especificamente os conteúdos de Convolucionais. Para isso, será selecionado um conjunto de dados, sobre o qual serão aplicadas técnicas de ajuste para o problema específico , além de técnicas de visualização (como histogramas) para, por fim, aplicar diversos algoritmos de classificação e comparar seus resultados.

Assim, mostrou-se possível exercitar os conceitos aprendidos em sala, de forma que o grupo obteve reflexões e ponderações sobre o funcionamento dos algoritmos e do processo de preparação dos dados.

## **2. Escolha do Conjunto de Dados**

O conjunto escolhido para a pesquisa é denominado *Sign Language MNIST*(<https://www.kaggle.com/datasets/datamunge/sign-language-mnist> ), e contém informações sobre a lingua de sinais americana, com cerca de 35 mil imagens - porém em forma de MNIST

A escolha deste conjunto se deve a presença de interesse dos membros em saber se é possível construir um tradutor competente, além da necessidade de tratamento dos atributos na medida em que os valores apresentados precisam entrar em consenso nas suas dimensões, mas também eu como esse tema é pertinente nos dias atuais, havendo séries e outros projetos que buscam esse interesse

O desafio proposto por esse conjunto de dados foi tentar identificar os possíveis sinais que aparecem, se baseando num conjunto que anteriormente teria sido reconhecido. Com isso, construir um possível tradutor para estes gestos.

O conjunto de dados contém 24 gêneros diferentes, (). Todas as manipulações dos dados e aplicações podem ser encontradas no notebook do Collab, que pode ser acessado nestes links: [Link da realização do trabalho - CPU](https://colab.research.google.com/drive/18jwuDnR84KEUIt2xgABwwfRKGgPDW9Un?authuser=1#scrollTo=Qi4r1HOpsePf). [GPU](https://colab.research.google.com/drive/1ffI8uNlIPLIiqBMDZ62KiZUOnzr_bmT7?authuser=1#scrollTo=ybvKtVDmYp9a)

## **3. Estudo do Conjunto de Dados**

### **3.1. Descrição do conjunto de dados**

Os dados foram coletados na tentativa de identificar os sinais que uma pessoa usa com base naqueles que já são conhecidos. Para isso, há dois datasets, um contendo dados de treino com 7 mil entradas, contendo de forma equilibrada todos os sinais, menos o “i”. Isso tudo dentro de um CSV onde contém os dados de cada imagem. Mas também outro semelhante, com 24 mil entradas, para testar.

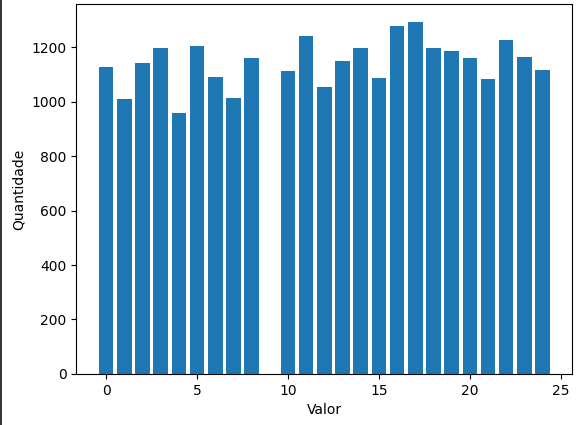
Esse dataset foi escolhido por alguns motivos, dentre eles foi a questão de explorar mais como funcionam as Redes Neurais de Convolução, já que há interesse de todos do grupo. Interesse de como processar, reconhecer e principalmente traduzir os sinais da linguagem de sinais em geral, visto que é um tema muito pertinente em todo o mundo. Uma das cenas da Marvel, da série “Echo” apresenta um tradutor funcional. Saber como seria difícil e desafiador criar um é um dos interesses.

Ademais, trata-se de uma CNN com multiclasse, sendo elas 24 letras. O conjunto é bem grande, mas também interessante. Além de haver vários sinais que são parecidos.

### **3.2. Atributos**

Abaixo, estão descritos os atributos utilizados para a classificação dos livros.

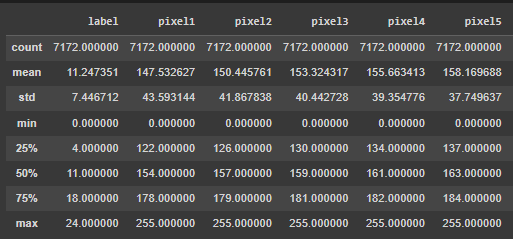
* **#Indíce**
* **Label**

****

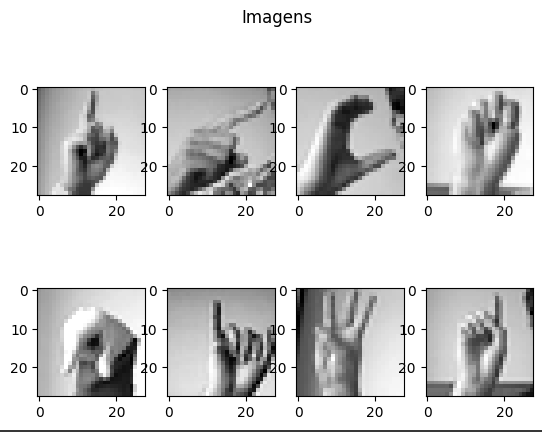
****

***Imagem 0 -*** *Labels existentes - Cada um representa uma letra*

* **Pixel** As demais colunas representam os dados das imagens presentes, com elas é possível construir certinho.



***Imagem 1 -*** *Informações dos Pixels*

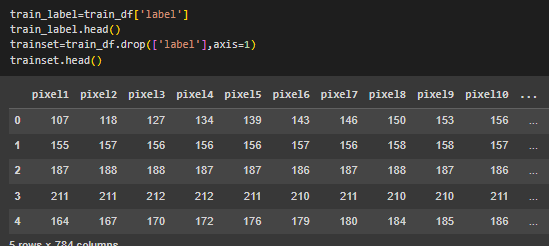
**

***Imagem 2 -*** *Reconstrução dos Pixels*

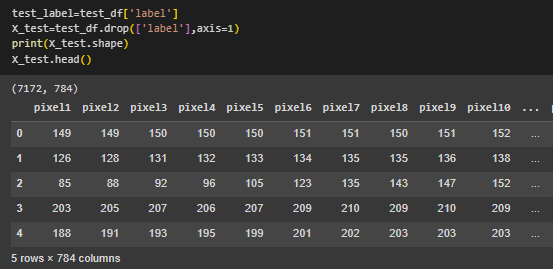
## **4. Tratamento de Dados**

Após essa série de visualizações, mais o cuidado em cada .csv fornecido. Percebeu-se que nos dados de treino, algumas letras não estavam presentes - havia somente 24 e não 25 conforme especificado no dataset - letra “j”. Mesmo assim, foi executado o trabalho desconsiderando esse fator, apenas vai perder uma delas. Em contraposição, na tabela dos csv não há valores faltantes - NaN - o que é uma boa informação.

Além desse cuidado, precisou realizar algumas mudanças, uma delas foi dividir a coluna de labels do pixel para os dados de treinamento. Dessa forma teremos os dados divididos de forma certa, mas também ideal para uma transformação futura. Essa transformação também foi vista nos dados de teste.

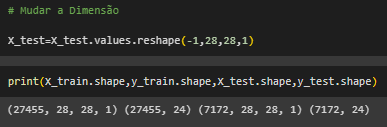


***Imagem 3 -****Preparando colunas x*



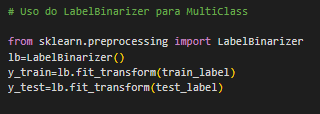
***Imagem 4 -****Dividindo colunas x*

Mas também, houve a necessidade de transformar as duas entradas de teste e treinamento para o “shape” ideal, uma vez que estavam em 1D e precisavam estar em 3D, com isso eles ficariam 28x28x1 ( Devidas dimensões das imagens ) e cada um com a respectiva quantidade.



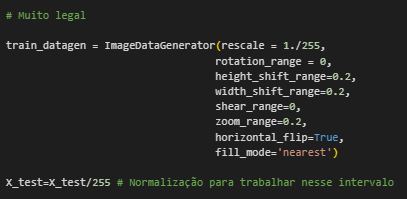
***Imagem 5 -****Colocando as dimensões para uso da CNN*

Por fim, das mudanças necessárias para trabalhar, lembrou-se que se trata de um problema MultiLabel, seria necessário escolher alguma estratégia,a qual foi dada ênfase a **LabelBinarizer()**que pega os rótulos e cria uma classe para cada um - feito para o teste e treinamento. Contudo, isso tem seus pontos negativos, visto que 24 colunas seriam adicionadas, mas também todos os atributos precisariam ser independentes - já que é uma abstração aceita ao utilizar esse método. Mas como são realmente diferente, não terá problema



***Imagem 6 -*** *Tratamento de Dado aplicado*

Por fim, o grupo decidiu de usar uma abordagem interessante para aumentar a dificuldade de classificação - já que as imagens são relativamente simples. Para tal, usou-se da função ImageDataGenerator() que pega os dados das imagens passados e usa a geração de certas constantes que servem para rotacionar, aumentar o zoom, diminuir, ou escalar imagens, pixel a pixel. Com isso, obtendo outras imagens com diferenciação, mas também escaladas e delimitadas.



***Imagem 7 -****Modificações nas imagens de treinamento*

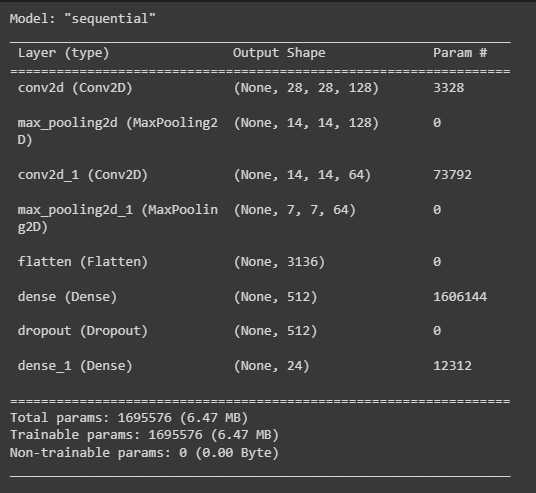
## **5. CNN.**

Com tudo isso feito, agora vem a parte de montar o classificador com as suas devidas características. A prior, foi pensado nas estratégias que poderiam ser usadas, declarando a quantidade de camadas ( seus respectivos kernels e tamanho - testar para ver o funcionamento das redes), poolings (tamanho), quantidade de unidades na camada densa e por fim a saída, e principalmente as funções de ativação ( testar praticamente os conhecimentos vistos em aula ). Dessa maneira foi pensando nas seguintes abordagens.

Importante ressaltar que as entradas vão ser todas de 28x28x1 para trabalhar com o 3D. Mas também, Quando padding='same', o número de zeros adicionados em torno da entrada é calculado automaticamente para garantir que o tamanho da saída seja o mesmo que o tamanho da entrada, dado o tamanho do filtro e o passo (strides) especificados. O objetivo é evitar a perda de informações nas bordas da imagem durante a aplicação das operações. Já o “strides” significa que a janela de pooling se moverá 2 unidades de passo a cada iteração.

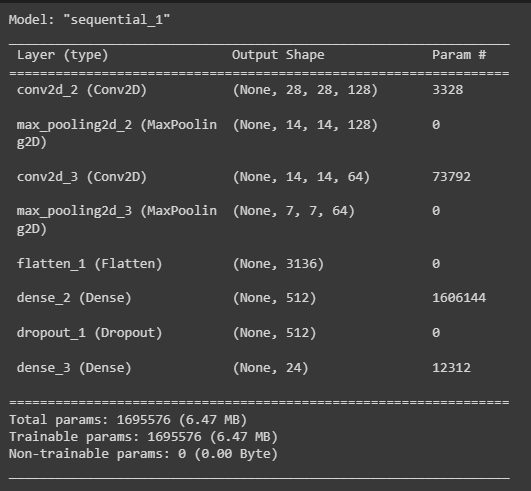
Ademais, o Pooling escolhido foi o Max Pooling, utilizada em redes neurais convolucionais (CNNs) para reduzir a dimensionalidade dos mapas de características, preservando as características mais importantes. Dropout com 25% para reduzir o overfitting. Por fim, uso da função softmax, pois transforma um vetor de valores reais em um vetor de probabilidades, onde cada elemento no vetor de saída representa a probabilidade de pertencer à classe correspondente. Isso é especialmente útil em problemas de classificação em que cada entrada pode pertencer a uma única classe.

| **2 Camadas - ReLu** | | | | | |
| --- | --- | --- | --- | --- | --- |
| Camadas | Função de Ativação | Pooling ( MaxPooling ) | Camada Densa | DropOut | Saída |
| 5x5, 128 padding = “same” | ReLu | pool\_size=(3,3)strides=2  padding=same | 512 Unidades  activation='relu' | 25% | 24 para 24 classes diferentes  activation='softmax' |
| 3x3, 64  padding = “same” | ReLu | (2,2),2,padding='same' |



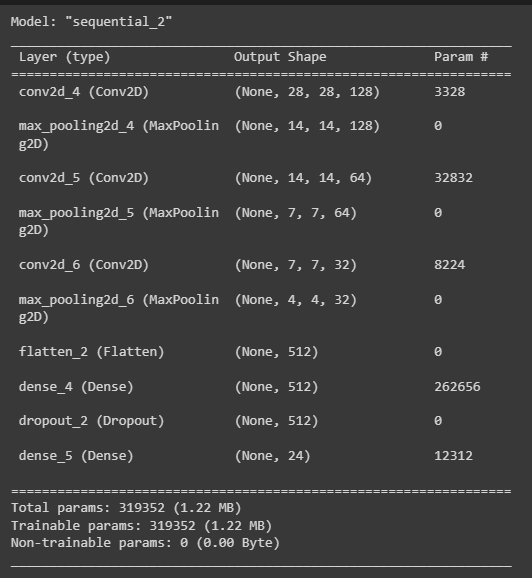
***Imagem 8 -****2 Camadas usando Relu*

| **2 Camadas - LeakyReLU** | | | | | |
| --- | --- | --- | --- | --- | --- |
| Camadas | Função de Ativação | Pooling ( MaxPooling ) | Camada Densa | DropOut | Saída |
| 5x5, 128 padding = “same” | LeakyReLU | pool\_size=(3,3)strides=2  padding=same | 512 Unidades  activation='relu' | 25% | 24 para 24 classes diferentes  activation='softmax' |
| 3x3, 64  padding = “same” | LeakyReLU | (2,2),2,padding='same' |



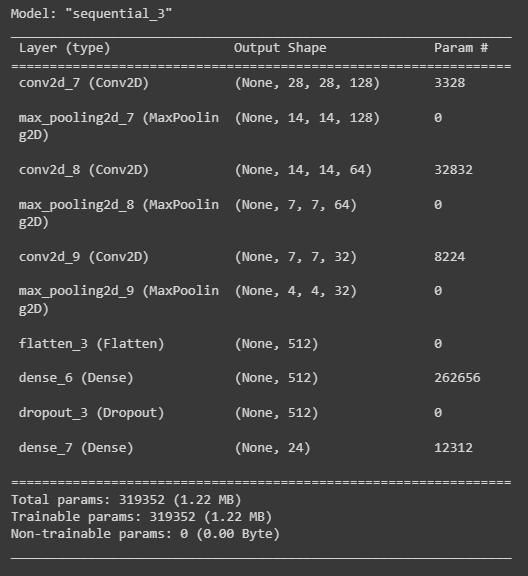
***Imagem 9 -****2 Camadas usando Leaky Relu*

| **3 Camadas - Relu** | | | | | |
| --- | --- | --- | --- | --- | --- |
| Camadas | Função de Ativação | Pooling ( MaxPooling ) | Camada Densa | DropOut | Saída |
| 5x5, 128 padding = “same” | ReLu | pool\_size=(3,3)strides=2  padding=same | 512 Unidades  activation='relu' | 25% | 24 para 24 classes diferentes  activation='softmax' |
| 2x2, 64  padding = “same” | ReLu | (2,2),2,padding='same' |
| 2x2, 32  padding = “same” | ReLu | (2,2),2,padding='same') |



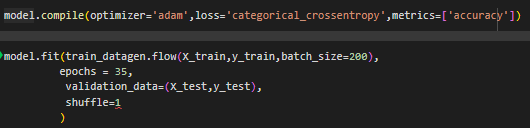
***Imagem 10 -*** *3 Camadas usando Relu*

| **3 Camadas - LeakyReLU** | | | | | |
| --- | --- | --- | --- | --- | --- |
| Camadas | Função de Ativação | Pooling ( MaxPooling ) | Camada Densa | DropOut | Saída |
| 5x5, 128 padding = “same” | LeakyReLU | pool\_size=(3,3)strides=2  padding=same | 512 Unidades  activation='relu' | 25% | 24 para 24 classes diferentes  activation='softmax' |
| 2x2, 64  padding = “same” | LeakyReLU | (2,2),2,padding='same' |
| 2x2, 32  padding = “same” | LeakyReLU | (2,2),2,padding='same') |



***Imagem 11 -****3 Camadas usando Leaky Relu*

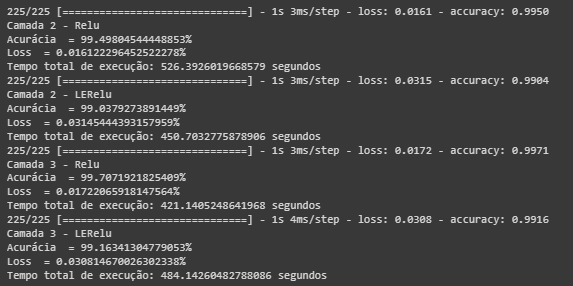
Com esses modelos prontos, foi preciso decidir alguns pontos finais. Para compilar os modelos foi usado o otimizador “adam” que é um otimizador bastante popular e eficiente que combina as ideias do Gradiente Descendente Estocástico (SGD) com adaptação de taxa de aprendizado. Mas também as duas métricas para avaliação: loss='categorical\_crossentropy', metrics=['accuracy'] que serão explicadas adiante. Por fim, em relação ao número de passos, para as de 2 camadas foi definido o valor de 40 e para as de 3 foi 35.



***Imagem 12 -*** *Exemplo de compilação*

## **6.Resultados**

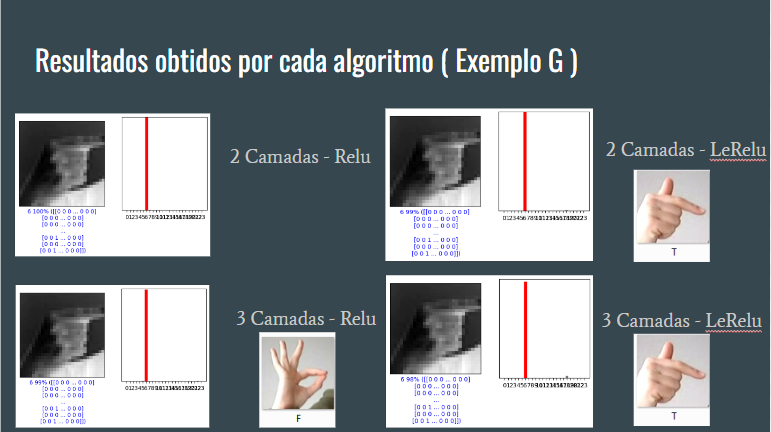
Os resultados da classificação multiclasse com cada método estão resumidos na tabela abaixo. Para avaliar o desempenho dos classificadores, utilizamos acurácia e loss. A acurácia é uma medida que considera apenas acertos inteiros o que, em nosso caso, corresponde a situações nas quais as classes foram previstas com sucesso ( probabilidade maior ) . Já o Loss verifica a questão de possibilidade, verificando se houve ou não acertos que com menos certeza, o que a torna uma medida relevante.



***Imagem 13 -****Métricas de Avaliação*

| **Modelo** | **Loss (0/1)** | **Acurácia (%)** | **Tempo CPU (s)** | **Tempo**  **GPU(s)** |
| --- | --- | --- | --- | --- |
| **2 Camadas - ReLu** | 0.0161 | 99.49 | 6716 | 526 |
| **2 Camadas - LeakyReLU** | 0.0315 | 99.03 | 7061 | 450 |
| **3 Camadas - Relu** | 0.0172 | 99.70 | 4303 | 421 |
| **3 Camadas - LeakyReLU** | 0.0308 | 99.16 | 4440 | 484 |

Verificamos que o algoritmo com melhor desempenho foi o 3 Camadas usando a ativação de Relu, comparando sua acurácia de 99,7 com os outros. Contudo, como podemos ver nas imagens abaixo, o Loss da de 2 Camadas é menor, o que garante mais acertos certeiros. Mas, como a diferença é pequena, e o tempo é menor, o de 3 Camas usando a Relu é melhor.



***Imagem 14 -*** *Teste para os modelos*

Essa imagem demonstra o loss menor das 2 Camas com o Relu, mas também das 3 Camas. Além de falar dos chutes de cada algoritmo.

## **7. Conclusão**

Após o estudo do conjunto de dados, foi possível produzir uma série de modelos de aprendizado de máquina capazes de prever o que cada gesto significa com certa eficácia. Dentre esses modelos, o que apresentou maior relevância, com as métricas de avaliação foi o 3 Camadas usando o ReLu, dado um Loss pequeno e uma acurácia significante, mas também pelo seu baixo valor de tempo de treino, dentre esses, este é o mais competente a ser utilizado na tradução de gestos. Contudo, como o objetivo era criar um tradutor em situações de cotidiano, é muito provável que isso não seja atingido.

Isso é devido a alguns detalhes, o primeiro é que o tempo computacional é bem elevado para treinar esses classificadores, numa situação real demandaria tempo o que dificultaria a comunicação entre as duas ou mais partes. Além disso, querendo ou não as imagens são relativamente simples, há sempre o mesmo fundo, mesma pessoa e mesmo esquema para tirar as fotos, o que faz com que o acerto seja elevado em casos simples, mas que talvez não seja tão bom quando houver redução da visibilidade - estar escuro, tom de pele se misturar com o ambiente etc. Por fim e mais fatal, há diversas situações que este tradutor não seria tão útil, visto que ele só trata as letras, é preciso trabalhar com palavras, números entre outras coisas, afinal se trata de uma língua e não somente de um alfabeto. Ou seja, ele não seria competente para o dia a dia, mas ajudaria em didáticas de ensino por letra, fugindo muito da cena vista na série “Echo”.

Adicionalmente, a realização deste trabalho, além de permitir revisar conceitos teóricos e suas aplicações práticas com a linguagem Python e suas bibliotecas, também permitiu o estudo de situações das redes neurais e funções de ativação vistas em sala de aula. Ademais, permitiu adentrar em um tema pertinente, que seria tradução de línguas visuais.

## **9. Bibliografia**

[https://www.kaggle.com/datasets/datamunge/sign-language-mnist](https://www.kaggle.com/datasets/ishikajohari/best-books-10k-multi-genre-data/data)

[CPU - Colab](https://colab.research.google.com/drive/18jwuDnR84KEUIt2xgABwwfRKGgPDW9Un?authuser=1#scrollTo=5YaDbS5fsXGU)

[GPU - Colab](https://colab.research.google.com/drive/1ffI8uNlIPLIiqBMDZ62KiZUOnzr_bmT7?authuser=1#scrollTo=ybvKtVDmYp9a)

[Pasta Drive](https://drive.google.com/drive/u/1/folders/11QvVvHWiCC8nqDaMw_EFHxFbRLrJ0ZKH)

]