

# Relatório de Entrega Técnica

Mini-Projeto 1 - Deep Learning e IA Generativa

**Nome do(a) Aluno(a):** Júlio Padilha, Nicole Victory, Roberto Arruda e Walter Barreto

**Título do Projeto:** Reconhecimento de Linguagem de Sinais (ASL) com CNN e Transfer Learning

**Data:** 13/04/2025

## 1. Temática e Dataset

Para o desenvolvimento do projeto e aplicação de uma CNN, a equipe decidiu por abordar um problema de reconhecimento de imagem para a ASL Alphabet (American Sign Language Alphabet), o alfabeto de em linguagem de sinais americano. O ASL é uma linguagem natural que serve como a principal língua de sinais das comunidades surdas nos Estados Unidos e na maior parte do Canadá anglófono. Ele é tradicionalmente composto por um conjunto de 26 sinais conhecido como o alfabeto manual americano, que pode ser usado para soletrar palavras do idioma inglês.

No dataset escolhido, trata-se de identificar 29 categorias de imagens: uma categoria para cada uma das 26 letras do alfabeto, uma categoria para “espaço”, uma categoria para representar “delete” e uma categoria para “nothing” em que não há sinais sendo representados na imagem.

O dataset escolhido pode ser encontrado no seguinte link: <https://www.kaggle.com/datasets/grassknotted/asl-alphabet/data>. Considerando que a base com 3.000 ocorrências de cada categoria, totalizando 78.000 exemplos. Dado o objetivo deste trabalho e por motivos de eficiência, a quantidade de exemplos foi reduzida para 500 exemplos categoria, totalizando 14.500 linhas no dataset utilizado.



Figura 1: Sample images do dataset ASL Alphabet

## 2. CNN e Busca por Hiperparâmetros

A abordagem para a construção de uma CNN inicial se baseou nos requisitos mínimos apresentados para o projeto: 3 redes convolucionais e 1 camada densa.

Para realizar a busca dos melhores hiperparâmetros para a rede a ser desenvolvida, a equipe escolheu adotar a metodologia do Random Search a partir do `keras_tuner`. O random search permite a execução do modelo com hiperparâmetros randômicos a partir de conjuntos de valores predeterminados pelo programador, essa abordagem se mostrou vantajosa ao grupo por apresentar uma melhor cobertura de opções em menor tempo. Este

fator se mostrou crucial para o time dada a quantidade de hiperparâmetros a serem testados e seu possíveis valores, além do longo tempo necessário para aprendizagem em cada vez que o modelo for treinado.

Os seguintes hiperparâmetros foram testados:

- *Quantidade de camadas convolucionais adicionais*: Foram testadas entre 3 a 6 camadas adicionais, abrangendo desde o mínimo requerido até o dobro desse valor, com o objetivo de explorar diferentes profundidades da rede e seu impacto na extração de características visuais.
- *Quantidade de poolings adicionais*: Avaliou-se de 0 a 2 camadas adicionais de pooling, partindo da mínima estrutura possível até um total de 3 camadas de pooling, permitindo investigar como o downsampling afeta a generalização e o tempo de treinamento.
- *Números de neurônios por camada densa*: Os valores 64, 100, 128 e 256 foram testados para cobrir desde arquiteturas mais leves (64 neurônios) até configurações mais expressivas e complexas (256), permitindo avaliar o impacto da capacidade de representação da rede no desempenho do modelo.
- *Quantidade de camadas densas intermediárias*: Foram avaliadas de 1 a 3 camadas para investigar o equilíbrio entre simplicidade e profundidade na fase de classificação, observando se camadas adicionais melhoraram a generalização ou introduzem sobreajuste.
- *Percentagem de dropout*: Os valores 0.25 e 0.5 foram utilizados por serem comumente adotados na literatura para reduzir o overfitting. O valor mais baixo preserva mais neurônios ativos, enquanto o mais alto força maior regularização.
- *Velocidade de aprendizagem*: Testes entre 0.0001 e 0.1 visaram identificar uma taxa que permita convergência eficiente sem comprometer a estabilidade. O intervalo cobre desde valores muito conservadores até taxas mais agressivas, adequando-se a diferentes arquiteturas e tamanhos de batch.

O Random Search foi executado em um total de 10 tentativas para a busca de valores mais apropriados. A quantidade de épocas foi fixada em 20, valor escolhido empiricamente com base em teste prévio dado que muitos dos modelos ainda se mostravam longe da convergência com o uso de 10 épocas (valor comumente utilizado para iniciar avaliação e construção de modelos). A acurácia foi definida como a métrica objetivo para identificar o melhor modelo, i.e. o modelo com maior acurácia seria definido como o possuidor do melhor conjunto de hiperparâmetros para determinar as classes com eficácia.

Após 10 tentativas, o modelo apresentando acurácia de 0.9429 no grupo de validação foi identificado com os seguintes parâmetros:

- 6 camadas de convolução adicional
- 2 poolings adicionais
- 256 neurônios por camada densa
- 1 camada densa intermediária
- 0.5 em dropout rate
- 0.0028074 de taxa de aprendizagem

### 3. Técnica de Regularização

O Dropout foi utilizado neste trabalho como uma técnica de regularização, atendendo ao requisito de usar pelo menos uma técnica de regularização, conforme solicitado. O Dropout é uma técnica amplamente usada para evitar o overfitting durante o treinamento de redes neurais. O mecanismo do Dropout consiste em desligar aleatoriamente uma porcentagem de neurônios em cada iteração de treinamento, impedindo que o modelo se torne excessivamente dependente de certas características dos dados e, assim, melhorando sua capacidade de generalização.

Para este trabalho, foi aplicada a técnica de Dropout antes da última camada densa do modelo, com valores de probabilidade de 0.25 e 0.5 para a desativação dos neurônios. No entanto, durante os experimentos, observou-se que a aplicação de Dropout teve um impacto limitado no desempenho geral do modelo.

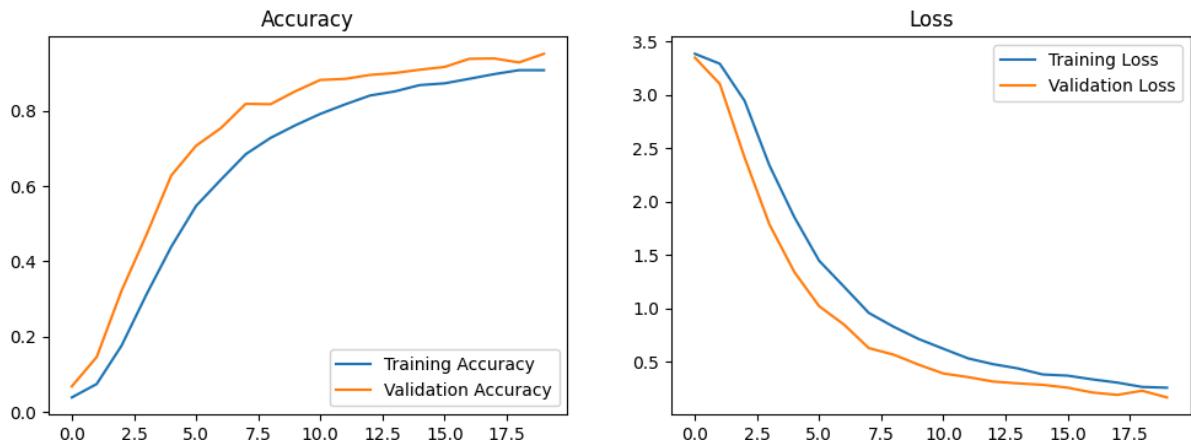
#### 3.1. Comparação dos Modelos

As métricas obtidas demonstram que a diferença de desempenho entre os dois modelos foi mínima.

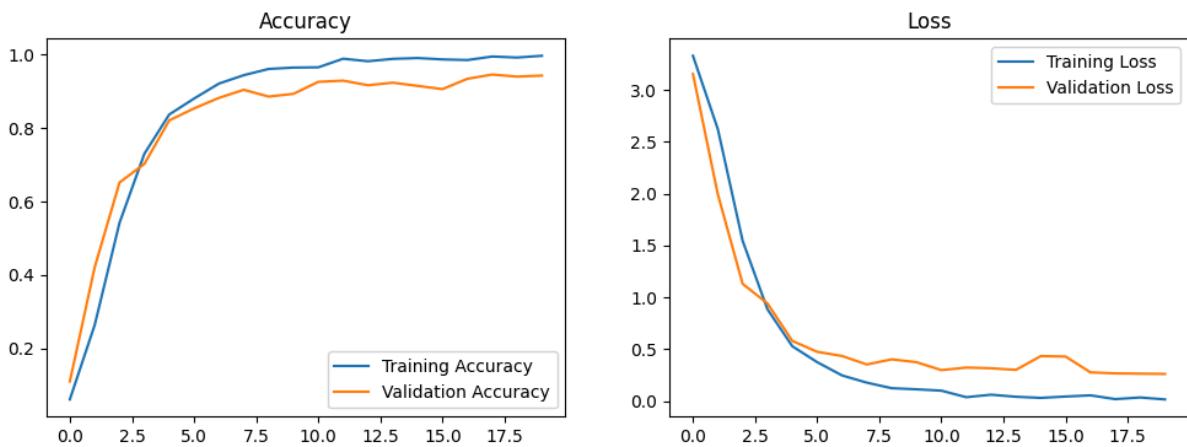
Modelo	Accuracy	Precision	Recall
Sem Dropout	0.9444	0.9461	0.9444
Com Dropout	0.9453	0.9461	0.9453

*Tabela 1: Comparativo de métricas entre modelos sem e com Dropout*

Além disso, ao observar os gráficos de curva de perda e acurácia ao longo das épocas, nota-se que ambos os modelos mantêm estabilidade durante o treinamento, sem oscilações abruptas ou sinais claros de overfitting, reforçando a ideia de que o Dropout, embora aplicado corretamente, não trouxe benefícios relevantes neste contexto específico.



*Figura 2 : Gráficos de curva de acurácia e perda ao longo das épocas do modelo sem dropout*



*Figura 3: Gráficos de curva de acurácia e perda ao longo das épocas do modelo com dropout*

### 3.2. Razões para o Impacto Limitado do Dropout

A razão para o efeito limitado do Dropout pode estar relacionada a algumas características específicas do nosso cenário de aplicação:

#### 3.2.1. Tamanho do Dataset:

O conjunto de dados ASL utilizado é relativamente grande, com várias imagens para cada classe. Como as imagens já possuem variações naturais devido à diferença de pose e contexto, a quantidade de dados foi suficiente para que o modelo aprendesse de maneira eficiente sem depender de uma regularização agressiva. Em datasets menores, a regularização tende a ser mais eficaz porque o modelo corre o risco de memorizar os dados (overfitting), mas isso não ocorreu com tanta intensidade neste caso devido ao tamanho da base de dados.

#### 3.2.2. Arquitetura do Modelo:

A rede neural foi construída de forma relativamente simples, com poucas camadas convolucionais e densas. Modelos mais profundos e complexos, que contêm um número elevado de parâmetros, tendem a sofrer mais de overfitting, e portanto, o Dropout seria mais impactante. Por outro lado, redes mais rasas, como a utilizada neste trabalho, podem ser menos suscetíveis a esse tipo de problema, já que o número de parâmetros é limitado.

## 4. Transfer Learning

Após o desenvolvimento de treinamento de um modelo por completo, a equipe deveria aplicar técnica do transfer learning objetivando utilizar um modelo previamente treinando e aplicar uma novo conjunto de camadas de redes neurais para sua aplicação no problema apresentado.

O modelo escolhido foi o VGG16, uma rede convolucional profunda composta por 16 camadas treináveis, conhecida por sua arquitetura simples e sequencial, com blocos de convoluções 3x3 e pooling 2x2, com os pesos produzidos a partir do treino com os dados da Imagenet. O modelo foi escolhido por ser um modelo amplamente utilizado para classificação de imagens, tendo sido observado seu uso em outras abordagens para classificação e identificação ASL, além de ser o modelo utilizado como exemplo nos materiais complementares do curso em questão.

Para este projeto, a equipe decidiu por realizar o transfer learning sem aplicação de fine tuning e utilizando a mesma quantidade de camadas densas, taxa de dropout e quantidade de neurônios por camadas densas (1 camada intermediária, 256 neurônios, 0.5 taxa de dropout). Tal abordagem foi escolhida com o objetivo de permitir ao time comparar o desempenho de um modelo com pesos pré-treinados por Transfer Learning com o de um modelo treinado do zero, analisando os impactos diretos dessa diferença na extração de características e nos resultados obtidos com o conjunto de dados específico.

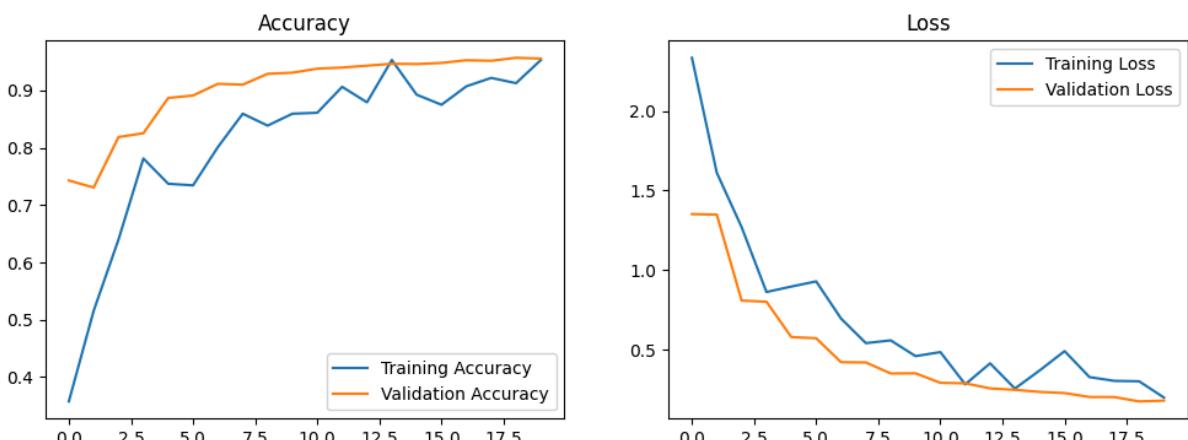
#### 4.1. Comparação dos Modelos

Realizando a avaliação dos modelos com e sem transfer learning (ambos com Dropout), assim como no comparativo de modelos com e sem técnicas de regularização, as métricas obtidas demonstram que a diferença de desempenho entre os dois modelos foi mínima.

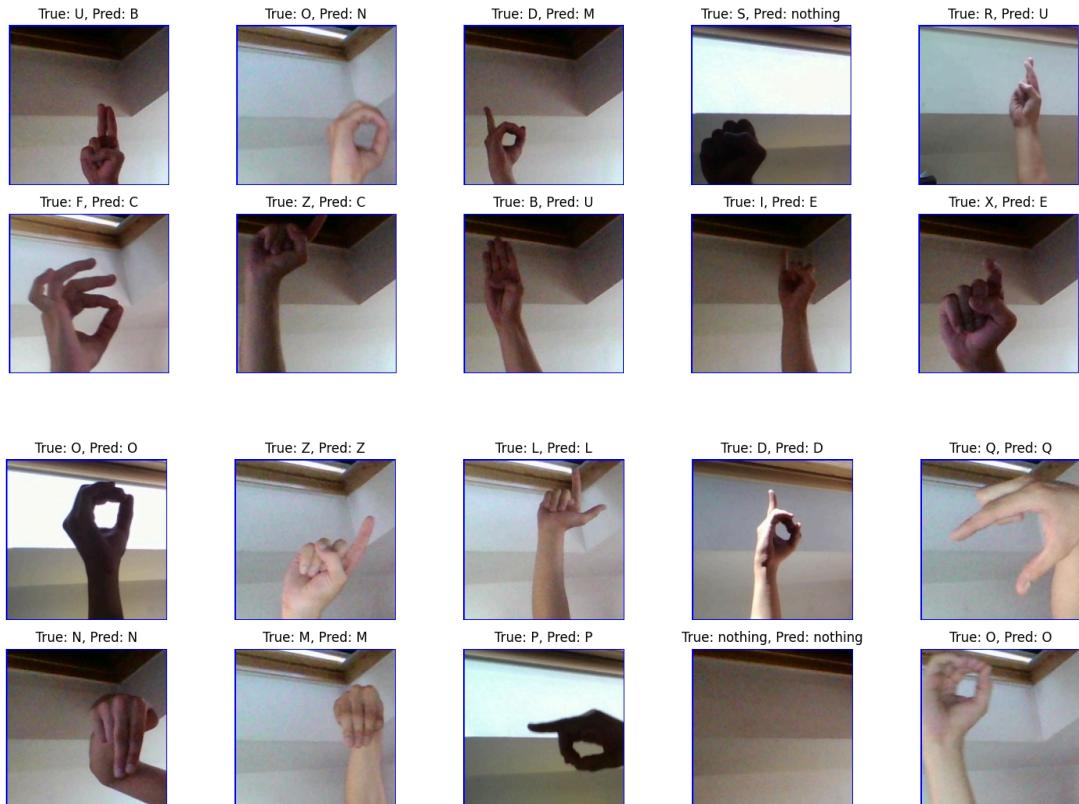
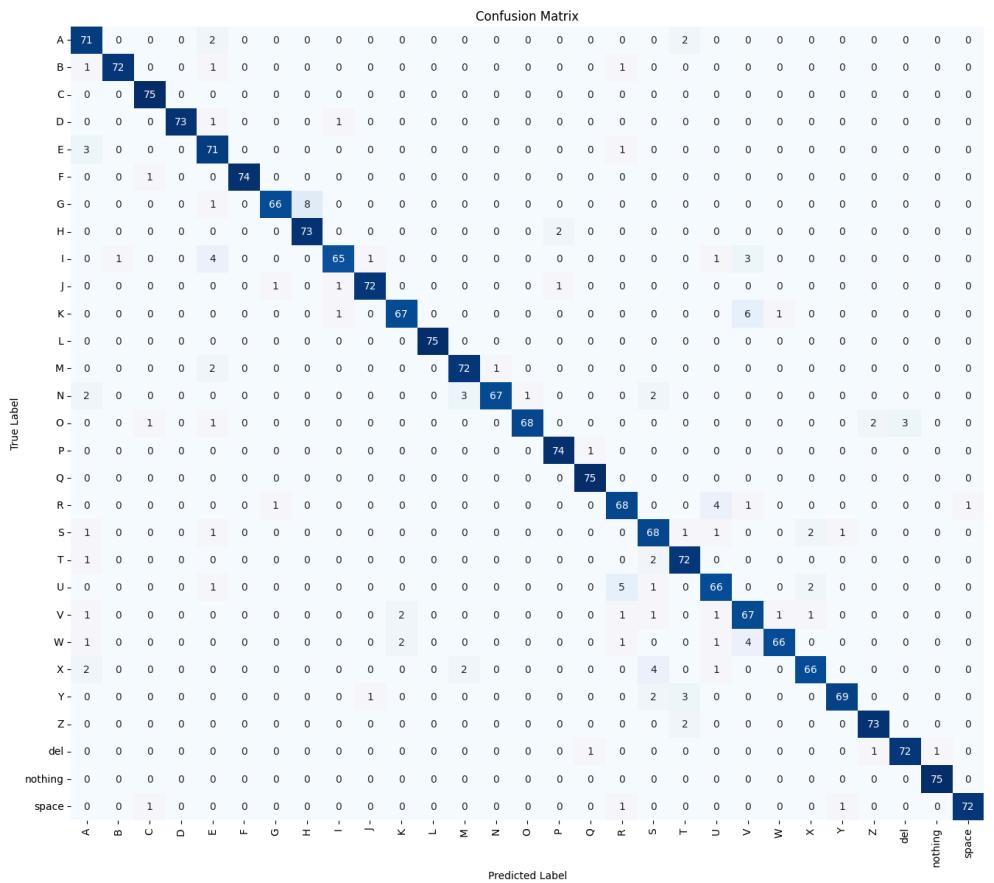
Modelo	Accuracy	Precision	Recall
Sem Transfer Learning	0.9453	0.9461	0.9453
Com Transfer Learning	0.9498	0.9526	0.9498

Tabela 2: Comparativo de métricas entre modelos sem e com Transfer Learning

Comparando os gráficos de histórico de acurácia e perda é possível perceber que assim como no modelo completamente treinado, houve um desenvolvimento constante dos valores sem uma grande discrepância entre os valores para o grupo de treino e o grupo de validação, demonstrando que o modelo estava de fato aprendendo e não memorizando os valores estudados (overfitting). Diferentemente do modelo construído, a progressão de épocas do modelo transferido não garantiu um aprendizado tão suave no decorrer das épocas com oscilações que levaram a uma convergência final.



*Figura 4: Gráficos de curva de acurácia e perda ao longo das épocas do modelo com Transfer Learning*



*Figura 5, 6 e 7: Matriz de confusão, exemplos de acerto e exemplos de erro do modelo treinando*

		Confusion Matrix																										
		True Label																										
		Predicted Label																										
A -	71	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
B -	0	74	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C -	0	0	75	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D -	0	0	1	73	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E -	2	2	0	0	67	0	0	0	0	0	0	0	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0
F -	0	0	0	0	0	75	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G -	0	0	0	0	0	0	70	4	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H -	0	0	0	0	0	2	73	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
I -	1	0	0	0	0	0	0	72	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
J -	0	0	0	0	0	0	0	0	75	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K -	0	0	0	0	0	0	0	0	0	72	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
L -	0	0	0	0	0	0	0	0	0	0	74	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
M -	1	0	0	0	0	0	0	0	0	0	73	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N -	0	0	0	0	0	0	0	0	0	0	0	13	62	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
O -	0	0	0	1	0	0	0	0	0	0	0	2	0	72	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P -	0	0	0	0	0	0	0	0	0	0	0	0	0	72	0	0	0	0	0	0	0	0	0	0	0	0	3	0
Q -	0	0	0	0	0	0	0	0	0	0	0	0	0	0	75	0	0	0	0	0	0	0	0	0	0	0	0	0
R -	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	68	0	0	0	3	0	0	1	0	0	0	0	0
S -	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	71	0	0	0	0	4	0	0	0	0	0	0	0
T -	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	71	0	0	0	0	0	0	0	0	0	0	0
U -	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	12	0	0	55	2	0	3	0	0	0	0
V -	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	4	64	2	2	0	0	0	0	0	
W -	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	74	0	0	0	0	0	0	0	
X -	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	2	0	0	0	69	0	0	0	0
Y -	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	72	0	0	0	0	0
Z -	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	75	0	0	0	0	0
del -	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	74	0	0	0	0
nothing -	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	73	0
space -	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A -																												
B -																												
C -																												
D -																												
E -																												
F -																												
G -																												
H -																												
I -																												
J -																												
K -																												
L -																												
M -																												
N -																												
O -																												
P -																												
Q -																												
R -																												
S -																												
T -																												
U -																												
V -																												
W -																												
X -																												
Y -																												
Z -																												
del -																												
nothing -																												
space -																												



*Figura 8, 9 e 10: Matriz de confusão, exemplos de acerto e exemplos de erro do modelo transferido*

Analizando ambos os resultados, é possível ver que não houveram grandes ganhos ou perdas com a troca de abordagens. Nos dois cenários é perceptível que imagens com menos contraste e/ou menos luminosidade formam mais presentes no casos de previsões erradas.

Observando as matrizes confusões, o modelo treinado apresenta seus erros de modo mais distribuído entre todas as classes além de uma baixa incidência. Já o modelo transferido apresenta uma incidência de erros mais concentrados em classes específicas, como trocar “N” por “M” e “U” por “R”, que pode ser proveniente da semelhança das silhuetas em sua sinalização.

## 5. Conclusão

O presente trabalho demonstrou, na prática, como diferentes técnicas e abordagens de aprendizado profundo podem ser aplicadas à tarefa de reconhecimento de linguagem de sinais americana (ASL) por meio de imagens. Através da construção de modelos baseados em redes neurais convolucionais (CNNs), combinadas com estratégias como Transfer Learning e regularização com Dropout foi possível alcançar resultados expressivos em termos de acurácia e capacidade de generalização.

Ao longo do processo, observou-se que decisões relacionadas ao ajuste fino de hiperparâmetro foram tão ou mais relevantes que a arquitetura do modelo, ao uso de modelos pré-treinados e a organização dos dados. Técnicas que, em outros contextos, teriam um papel mais determinante, como o Dropout, apresentaram impacto discreto, sugerindo que nem sempre a regularização precisa ser agressiva quando o pipeline de treinamento já é bem estruturado.

Além dos resultados numéricos, o projeto também proporcionou um aprendizado em relação à construção experimental de modelos de visão computacional, reforçando a importância de testes controlados, visualização e análise criteriosa das curvas de desempenho para uma compreensão mais profunda do comportamento do modelo.

A construção do projeto foi capaz de mostrar na prática as vantagens do transfer learning. Sem grandes discrepâncias nas métricas alcançadas, o uso de um modelo pré-treinado economiza recursos e garante resultados semelhantes com menos tempo investido em seu aprendizado.

Por fim, os experimentos confirmam que, com uma combinação equilibrada de práticas modernas de treinamento, é possível construir soluções eficazes mesmo com recursos limitados, o que abre espaço para aplicações acessíveis e de grande impacto social no reconhecimento de sinais visuais como os do ASL. Como continuidade futura, sugere-se explorar métodos mais robustos de validação cruzada, ampliar o escopo para outras línguas de sinais e avaliar a aplicação em tempo real. Com isso, é possível transformar esse tipo de pesquisa em soluções mais acessíveis e inclusivas no campo da comunicação assistiva.

## 6. Referências

- [1] Wikipedia, American Sign Language, Wikipedia. [Online]. Available: [https://en.wikipedia.org/wiki/American\\_Sign\\_Language](https://en.wikipedia.org/wiki/American_Sign_Language). [Accessed: Apr. 13, 2025].
- [2] A. Nagaraj, *ASL Alphabet*. Kaggle, 2018. [Online]. Available: <https://www.kaggle.com/dsv/29550>. DOI: 10.34740/KAGGLE/DSV/29550. [Accessed: Apr. 03, 2025].
- [3] Keras, *Getting started with KerasTuner*, Keras Documentation. [Online]. Available: [https://keras.io/keras\\_tuner/getting\\_started/](https://keras.io/keras_tuner/getting_started/). [Accessed: Apr. 10, 2025].
- [4] TensorFlow, *tf.keras.layers.Dropout*, TensorFlow API Documentation. [Online]. Available: [https://www.tensorflow.org/api\\_docs/python/tf/keras/layers/Dropout](https://www.tensorflow.org/api_docs/python/tf/keras/layers/Dropout). [Accessed: Apr. 11, 2025].
- [5] My Great Learning, *Everything You Need to Know About VGG16*, Medium, Sep. 20, 2021. [Online]. Available: <https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918>. [Accessed: Apr. 11, 2025].