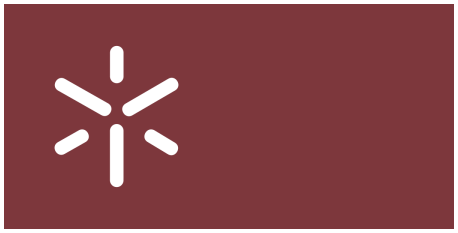

Universidade do Minho
Escola de Engenharia
Mestrado em Engenharia Informática

Unidade Curricular de Visão por Computador e Processamento de Imagem

Ano Letivo de 2024/2025

Visão por Computador - Trabalho de grupo

Francisco Afonso, PG57873
Jéssica Cunha, A100901
Martim Redondo, PG57889

2 de junho de 2025

Índice

1. Introdução

Este relatório descreve o processo de desenvolvimento e avaliação de modelos de *deep learning* aplicados ao dataset GTSRB (German Traffic Sign Recognition Benchmark), que consiste em imagens de sinais de trânsito alemães. O objetivo principal foi alcançar a melhor *accuracy* possível no conjunto de teste, aproximando-se do estado da arte publicado de 99.82%.

O trabalho foi dividido em duas fases principais: na primeira, exploramos diferentes técnicas de *data augmentation* e o seu impacto no desempenho dos modelos; na segunda fase, investigamos o potencial de usar *ensembles* das redes treinadas anteriormente.

2. Metodologia e desenvolvimento

2.1. Primeira Fase: Exploração de *Data Augmentation*

A primeira etapa concentrou-se na investigação do impacto de diferentes estratégias de aumento de dados no desempenho da classificação. O *data augmentation* é fundamental em visão de computadores, pois permite aumentar artificialmente o tamanho do conjunto de treino através de transformações às imagens originais, melhorando a capacidade de generalização dos modelos. Para esta investigação, foram desenvolvidas seis configurações distintas de aumento de dados, cada uma focada em aspetos específicos das variações presentes no *dataset*:

- **Configuração *default*:** Serviu como *baseline* incluindo transformações standard como redimensionamento para 32×32 píxeis, *flip* horizontal aleatório, rotação até 15 graus, ajustes de cor controlados e pequenas transformações afins.
- **Configuração *basic*:** Versão minimalista contendo apenas redimensionamento e *flip* horizontal, permitindo avaliar o impacto das transformações mais simples.
- **Configuração *geometric*:** Enfatizou transformações espaciais com rotações de até 30 graus, transformações afins intensificadas e distorções de perspetiva para simular diferentes ângulos de captura.
- **Configuração de *color*:** Especializada em variações de aparência visual com ajustes intensificados de brilho, contraste, saturação e matiz, conversão ocasional para escala de cinza e variações de nitidez. Esta abordagem visou preparar o modelo para diferentes condições de iluminação e estados de conservação dos sinais.
- **Configuração *agressive*:** Combinou todas as técnicas anteriores com parâmetros extremos, testando os limites do aumento de dados com rotações de até 45 graus e transformações muito pronunciadas.
- **Configuração *advanced*:** Apresentou uma abordagem equilibrada incorporando *Random Erasing*, uma técnica moderna que oculta aleatoriamente regiões da imagem para simular oclusões e forçar o modelo a aprender características distribuídas.

Para avaliar estas configurações, foi desenvolvida uma arquitetura CNN moderna (**ModernCNN**) que serviu como *baseline*. Esta arquitetura incluiu quatro blocos convolucionais com progressão de filtros de 32 para 256, intercalados com normalização por lotes, *dropout* e *max-pooling*, seguidos por camadas totalmente conectadas. O design ofereceu um equilíbrio adequado entre capacidade de aprendizagem e eficiência computacional.

2.2. Resultados da Primeira Fase

Cada configuração foi analisada visualmente para perceber o seu efeito nas imagens de treino.

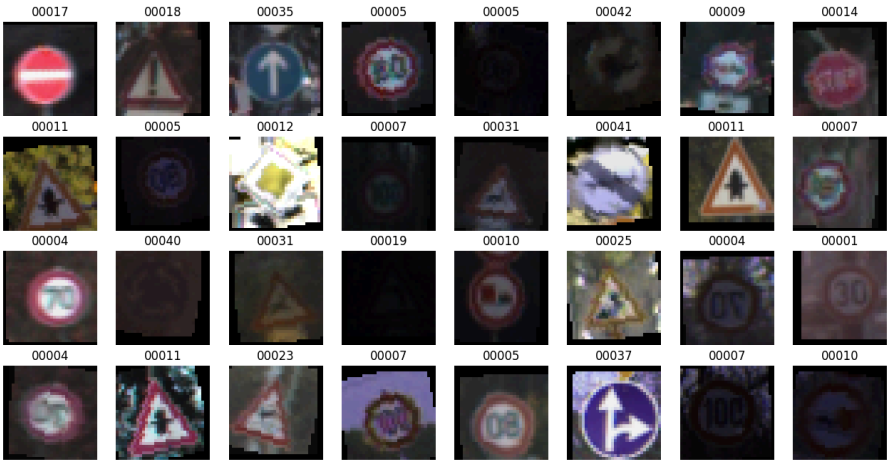

Figura 1: Imagens de treino após configuração *default*

As imagens mostram transformações moderadas que mantêm os sinais bem legíveis. Esta configuração oferece um bom equilíbrio entre diversidade dos dados e preservação da qualidade original das imagens.

Figura 2: Imagens de treino após configuração *basic*

As imagens mantêm-se muito próximas do original, com apenas inversões horizontais ocasionais. Apesar da simplicidade, esta abordagem mostrou-se eficaz, demonstrando que transformações básicas podem ser suficientes.

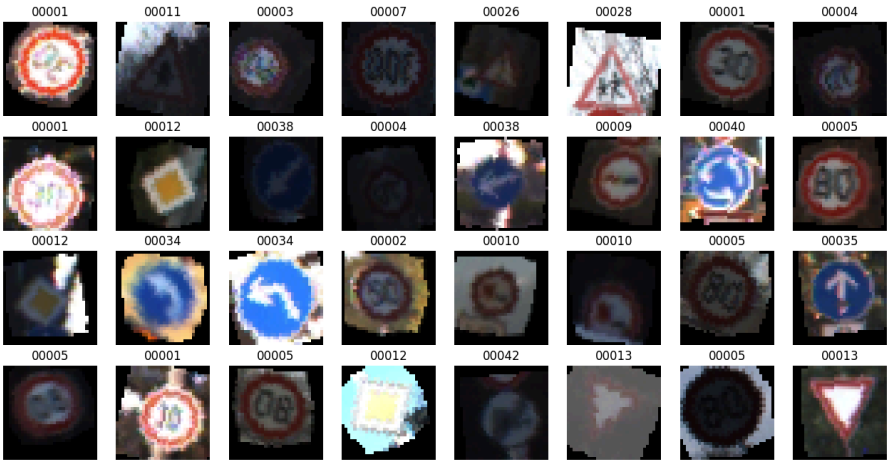

Figura 3: Imagens de treino após configuração *geometric*

As imagens apresentam rotações mais acentuadas e algumas distorções que simulam diferentes ângulos de captura. Estas transformações preparam o modelo para sinais vistos de diferentes perspetivas.

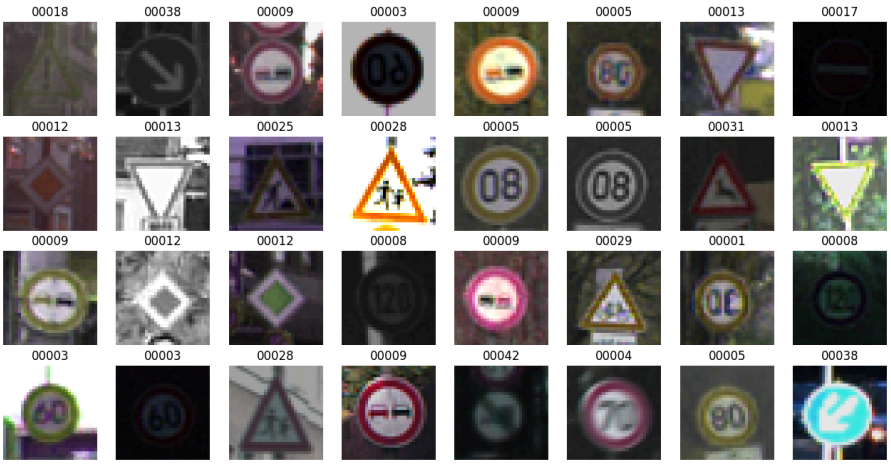

Figura 4: Imagens de treino após configuração *color*

As imagens mostram grande variedade de aparências, simulando diferentes condições de iluminação e estados de conservação dos sinais. Esta diversidade visual revelou-se a mais eficaz para melhorar o desempenho dos modelos.

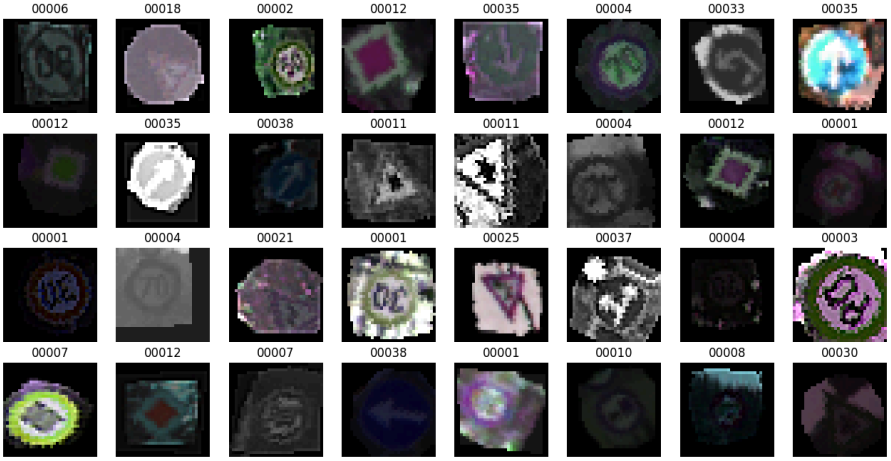

Figura 5: Imagens de treino após configuração *aggressive*

As imagens apresentam transformações muito intensas que por vezes dificultam a identificação visual dos sinais. Esta abordagem extrema acabou por prejudicar o desempenho dos modelos.

Figura 6: Imagens de treino após configuração *advanced*

As imagens mostram regiões ocultadas que simulam sombras ou obstruções reais. Esta técnica força o modelo a aprender características distribuídas por toda a imagem, melhorando a robustez.

2.3. Segunda Fase: Arquiteturas Avançadas

A segunda fase explorou arquiteturas de redes neurais mais sofisticadas. Foram desenvolvidos quatro modelos distintos, cada um incorporando técnicas arquiteturais diferentes para avaliar o seu impacto no desempenho.

2.3.1. ImprovedCNN - Blocos Residuais

O **ImprovedCNN** incorporou blocos residuais inspirados na arquitetura *ResNet*, que permitiram treinar uma rede mais profunda sem o problema de *vanishing gradients*. Cada bloco residual incluiu conexões de salto (*skip connections*) que contornam uma ou

mais camadas, permitindo que o gradiente flua diretamente durante o treinamento. A rede começou com uma camada convolucional 7×7 para capturar características iniciais amplas, seguida por três estágios de blocos residuais com número crescente de filtros (de 64 para 512). No final, uma camada de *Global Average Pooling* substituiu as camadas totalmente conectadas tradicionais, reduzindo o número de parâmetros e evitando *overfitting*. Essa estrutura foi eficaz para aprender padrões hierárquicos complexos, como detalhes finos em sinais de trânsito com formas semelhantes.

2.3.2. EfficientCNN - Convoluções Otimizadas

O **EfficientCNN** adotou convoluções *depthwise* separáveis para melhorar a eficiência computacional sem sacrificar a precisão. Em vez de aplicar convoluções tradicionais, que processam simultaneamente espaços e canais, esta arquitetura dividiu o processo em duas etapas: uma convolução espacial (por canal) seguida de uma convolução 1×1 para combinar os canais. Além disso, mecanismos de *squeeze-and-excitation* foram adicionados para recalibrar dinamicamente a importância de cada canal de características. Isso permitiu que o modelo priorizasse automaticamente os canais mais informativos, como aqueles que detetam cores ou símbolos distintivos nos sinais. Apesar de ter menos parâmetros que os outros modelos, o EfficientCNN manteve uma boa precisão, sendo ideal para aplicações que exigem baixo consumo de recursos.

2.3.3. AttentionCNN - Mecanismos de Atenção

O **AttentionCNN** usou mecanismos de atenção para ajudar a rede a concentrar-se nas partes mais importantes da imagem. Primeiro, a imagem passou por camadas convolucionais tradicionais, que extraem diferentes tipos de características, como bordas, formas e texturas. Em seguida, uma sub-rede de atenção analisou essas características e criou mapas que mostram quais regiões da imagem são mais relevantes. Esses mapas foram usados para dar mais peso às partes importantes da imagem antes da etapa de classificação final. Essa abordagem mostrou-se especialmente útil no reconhecimento de sinais de trânsito, onde detalhes como símbolos ou contornos costumam ter mais valor para identificar corretamente o sinal.

2.3.4. ModernCNN (Baseline)

A **ModernCNN** serviu como *baseline* e foi projetada para equilibrar simplicidade e desempenho. Composta por quatro blocos convolucionais com filtros crescentes (32 a 256), cada bloco incluiu normalização por lotes (BatchNorm), ativação ReLU e *max-pooling* para reduzir dimensionalidade. Camadas de *dropout* foram adicionadas para evitar *overfitting*. A rede terminou com camadas totalmente conectadas para classificação. Embora menos complexa que as outras arquiteturas, essa estrutura mostrou-se robusta para tarefas básicas de reconhecimento, especialmente quando combinada com técnicas de *data augmentation* focadas em cor (como na configuração *color*).

2.4. Resultados das Arquiteturas

Cada arquitetura foi treinada usando a configuração de aumento de dados de cor por 30 *epochs* com o otimizador *AdamW* e taxa de aprendizado adaptativa. Os resultados demonstraram características distintas de cada abordagem:

Accuracy (%) /Arquitetura	**default**	**basic**	**geometric**	**color**	**aggressive**	**advanced**
AttentionCNN	96.10	97.06	96.27	97.69	93.29	97.77
ImprovedCNN	92.06	94.35	92.53	94.87	89.75	95.20
EfficientCNN	92.06	94.58	93.35	95.77	89.05	95.76
ModernCNN	96.04	97.58	96.49	97.93	93.15	97.57

Tabela 1: Accuracy por arquitetura

O **AttentionCNN** (97.77%) teve o melhor desempenho porque consegue focar nas partes mais importantes da imagem. Para sinais de trânsito, isto significa dar mais atenção aos símbolos centrais e números, ignorando o fundo menos relevante.

A **ModernCNN** (97.93%) mostrou excelente desempenho como *baseline*, demonstrando que uma arquitetura bem equilibrada pode alcançar resultados muito competitivos. A combinação de normalização por lotes, *dropout* e estrutura progressiva de filtros revelou-se eficaz para esta tarefa.

O **EfficientCNN** (95.77%) teve *accuracy* ligeiramente menor, mas é o mais eficiente computacionalmente. Esta arquitetura usa menos recursos mantendo boa performance, sendo ideal para aplicações em tempo real ou dispositivos com limitações de processamento.

O **ImprovedCNN** (94.87%) mostrou que os blocos residuais, embora úteis para redes muito profundas, podem não ser necessários para este problema específico. As conexões residuais são mais vantajosas em arquiteturas com dezenas ou centenas de camadas, mas para esta tarefa, arquiteturas mais simples demonstraram ser suficientes.

3. Estratégias de Ensemble

Para maximizar o desempenho, foram investigadas estratégias de *ensemble* combinando as previsões dos melhores modelos. Duas abordagens foram implementadas:

- **Ensemble por Votação:** Combinação das previsões discretas através de votação majoritária entre os modelos componentes.

Configuração	**Accuracy (%)**
Advanced	97.03
Aggressive	92.26
Color	97.05
Basic	96.63
Default	94.89
Geometric	95.46

Tabela 2: Accuracy por configuração

O melhor resultado de ensemble foi obtido com a configuração color (97.05%), seguido de perto pela configuração advanced (97.03%). A votação majoritária mostrou-se menos eficaz que a média ponderada, pois perde informação importante sobre o quão certo cada modelo está da sua previsão. Isto pode levar a decisões menos precisas quando existe incerteza entre classes similares.

4. Conclusões

Este trabalho explorou técnicas de aumento de dados e arquiteturas avançadas de redes neurais para o reconhecimento de sinais de trânsito no *dataset GTSRB*. Os resultados demonstraram que transformações focadas em cor e aparência visual (como ajustes de brilho, contraste e saturação) foram mais eficazes do que abordagens puramente geométricas ou agressivas, alcançando uma *accuracy* de até 97.93% com a ModernCNN. Além disso, mecanismos de atenção, como os implementados no AttentionCNN, destacaram-se pela sua capacidade de focar em regiões relevantes da imagem, como símbolos e contornos, contribuindo para um melhor desempenho.

Apesar dos bons resultados, o desempenho final de 97.05% com estratégias de *ensemble* ainda ficou muito abaixo do melhor resultado publicado (99.82%), indicando espaço para melhorias. Limitações como imagens de baixa resolução, oclusões e condições de iluminação extremas foram identificadas como os principais desafios. Esses problemas sugerem a necessidade de técnicas adicionais, como super-resolução ou *transfer learning* com modelos pré-treinados em *datasets* maiores, para melhorar a robustez do sistema.