Visão por Computador e Processamento de Imagem

Mestrado em Engenharia Informática
Universidade do Minho

Trabalho Prático 2

Grupo 15

Trabalho realizado por:	*Número*
Miguel Velho Raposo	A94942
André Lucas Silva Verdelho	PG50221

Braga, 29 de maio de 2024

Conteúdo

1. Introdução

O reconhecimento de sinais de trânsito desempenha um papel crucial no desenvolvimento de sistemas de condução autónoma e de assistência ao motorista, contribuindo significativamente para a segurança e a eficiência no tráfego rodoviário. O German Traffic Sign Recognition Benchmark (GTSRB) é um dataset de referência amplamente utilizado para a criação e avaliação de algoritmos de reconhecimento de sinais de trânsito, devido à sua abrangência e complexidade.

Neste trabalho, exploramos técnicas avançadas de deep learning aplicadas ao dataset GTSRB, com o objetivo de atingir a maior precisão possível na classificação de sinais de trânsito. A tarefa de reconhecimento de sinais de trânsito apresenta desafios substanciais, como variações nas condições de iluminação, oclusões, e diferentes ângulos de visualização. Para enfrentar esses desafios, a aplicação de técnicas de data augmentation é essencial, aumentando a diversidade do conjunto de dados de treino e melhorando a capacidade dos modelos se generalizarem para novos dados.

O trabalho é dividido em duas partes principais. Na primeira parte, treinamos múltiplos modelos de deep learning, aplicando várias técnicas de data augmentation tanto em pré-processamento quanto de forma dinâmica durante o treino. O objetivo é investigar o impacto de cada técnica no desempenho dos modelos, medido principalmente em termos de precisão. Esta fase inclui a exploração de diversos métodos de processamento de imagem e a avaliação sistemática dos seus efeitos na performance das redes neuronais.

Na segunda parte, estudamos o potencial da utilização de ensembles de redes neuronais, combinando os modelos treinados na primeira fase para formar um ensemble. Ensembles de modelos podem capturar uma maior diversidade de padrões nos dados, potencialmente resultando em melhorias significativas na precisão de reconhecimento. Avaliamos o desempenho dos ensembles e comparamos com os resultados dos modelos individuais.

Além do desenvolvimento e treino dos modelos, este trabalho inclui uma análise detalhada dos resultados, destacando as técnicas de data augmentation que mais contribuíram para a melhoria do desempenho. O relatório final documentará todas as escolhas metodológicas, os procedimentos adotados, os testes realizados e os resultados obtidos, proporcionando uma visão compreensiva do processo e das conclusões alcançadas.

Através deste estudo, não procuramos apenas atingir altos níveis de precisão na classificação de sinais de trânsito, mas também contribuir para o avanço do conhecimento sobre as melhores práticas em data augmentation e o uso de ensembles no contexto de reconhecimento de sinais de trânsito. Este trabalho visa, portanto, tanto a melhoria prática dos sistemas de reconhecimento de sinais quanto o enriquecimento da literatura académica sobre técnicas de deep learning aplicadas a problemas complexos de visão computacional.

2. Modelo

2.1 Estrutura da Rede

A rede utilizada é uma arquitetura convolucional composta por múltiplas camadas, especificamente desenvolvida para o reconhecimento de sinais de trânsito no dataset GTSRB. Abaixo, apresentamos as principais características desta rede:

- **Camada Convolucional 1**:
  - `Conv2d`: Aplica uma convolução 2D com 3 canais de entrada (imagens RGB) e 128 canais de saída. O kernel size é 5.
  - `BatchNorm2d`: Normalização em lote para 128 canais.
  - `LeakyReLU`: Função de ativação LeakyReLU com um fator de leak de 0.01.
  - `Dropout2d`: Regularização com uma taxa de dropout de 25%.
- **Camada Convolucional 2**:
  - `Conv2d`: Convolução 2D com 128 canais de entrada e 256 canais de saída. O kernel size é 5.
  - `BatchNorm2d`: Normalização em lote para 256 canais.
  - `LeakyReLU`: Função de ativação LeakyReLU com um fator de leak de 0.01.
  - `MaxPool2d`: Subamostragem com uma janela de 2x2.
  - `Dropout2d`: Regularização com uma taxa de dropout de 25%.
- **Camada Convolucional 3**:
  - `Conv2d`: Convolução 2D com 256 canais de entrada e 512 canais de saída. O kernel size é 5.
  - `BatchNorm2d`: Normalização em lote para 512 canais.
  - `LeakyReLU`: Função de ativação LeakyReLU com um fator de leak de 0.01.
  - `MaxPool2d`: Subamostragem com uma janela de 2x2.
  - `Dropout2d`: Regularização com uma taxa de dropout de 25%.
- **Camada Totalmente Conectada 1**:
  - `Linear`: Camada totalmente conectada com 512*4*4 (8192) unidades de entrada e 128 unidades de saída.
  - `LeakyReLU`: Função de ativação LeakyReLU com um fator de leak de 0.01.
  - `Dropout`: Regularização com uma taxa de dropout de 20%.
- **Camada Totalmente Conectada 2**:

– `Linear`: Camada totalmente conectada com 128 unidades de entrada e `num_classes` unidades de saída (43 classes no caso do GTSRB).

A escolha desta rede para o reconhecimento de sinais de trânsito justifica-se pela sua eficácia comprovada em tarefas de visão por computador, especialmente na deteção e classificação de objetos em imagens. Esta arquitetura, composta por camadas convolucionais seguidas de camadas totalmente conectadas, permite a extração de características complexas dos dados, enquanto a utilização de técnicas como normalização em lots e dropout ajuda a evitar o overfitting e promover uma melhor generalização para novos exemplos.

2.2 Fluxo de Dados

O fluxo de dados através da rede convolucional é estruturado de forma a permitir a extração progressiva de características das imagens de entrada, seguido por uma transformação linear para realizar a classificação. Abaixo, descrevemos as etapas principais deste fluxo:

1. As imagens de entrada passam pela primeira camada convolucional, onde são aplicadas convolução, normalização, ativação e dropout.
2. O mesmo processo é repetido nas camadas convolucionais subsequentes, com a adição de max pooling após a segunda e terceira camadas convolucionais.
3. Após as camadas convolucionais, os dados são achatados e passam pelas camadas totalmente conectadas, que aplicam uma transformação linear, ativação e dropout.
4. Finalmente, a última camada totalmente conectada produz a saída com o número de classes correspondente ao problema de classificação.

3. Data Augmentation

No conjunto de dados GTSRB, há um desequilíbrio significativo entre as classes. Por exemplo, as classes 0, 19 e 32 possuem muito poucas imagens, cerca de 60 cada, enquanto outras classes, como 2 e 38, têm aproximadamente 750 imagens. Esse desequilíbrio pode levar a um desempenho não satisfatório dos modelos de classificação, pois o modelo pode tender a se concentrar mais nas classes com maior número de exemplos, negligenciando as classes minoritárias. Para mitigar esse problema, podemos aplicar algumas técnicas para melhorar a distribuição dos dados:

- **Aumento de Dados**: Podemos aumentar o número de exemplos das classes com menos imagens aplicando técnicas como rotação, alteração de cores e recorte das imagens. Isso nos permite gerar mais dados a partir dos exemplos existentes, equilibrando melhor o número de imagens entre as diferentes classes.
- **Aplicação de Pesos às Classes na Função de Perda**: Outra abordagem é ajustar a função de perda para penalizar mais fortemente o modelo quando ele comete erros nas classes com menos exemplos. Isso pode ser feito aplicando pesos às classes, fazendo com que o modelo aprenda a dar mais atenção às classes minoritárias durante o treino.

Essas técnicas podem ajudar a melhorar o desempenho do modelo, garantindo que ele seja mais robusto e tenha uma melhor capacidade de generalização, especialmente em relação às classes com menos dados.

3.1 Modelo I

Para compreender melhor o conjunto de dados, decidimos treinar o modelo sem aplicar qualquer técnica de data augmentation e examinar as instâncias em que ele classificou incorretamente as imagens do conjunto de teste.

O índice de precisão resultante foi de 0.9780, o que indica que o modelo, utilizando apenas o conjunto de treino original, conseguiu identificar corretamente 97.80% das imagens do conjunto de teste.

Durante a análise das imagens classificadas incorretamente, notamos que muitas delas apresentavam sinais de trânsito inclinados, com pouca luminosidade ou fora do centro da imagem, algumas estavam ligeiramente cortadas e outras tinham sinais descoloridos. Além disso, observamos que ocorreram muitos erros na classificação das velocidades e que uma parcela significativa das imagens apresentava um contraste bastante baixo, o que possivelmente dificultou a identificação das bordas.

3.2 Modelo II

Para melhor avaliar o desempenho do modelo e evitar overfitting, treinamos o Modelo II com a inclusão de um conjunto de validação. Todas as outras configurações e arquitetura da rede foram mantidas iguais ao Modelo I.

Após o treino, o Modelo II alcançou uma precisão de 97.17%, ligeiramente menor que o Modelo I. A introdução do conjunto de validação pode ter ajudado a regularizar o modelo.

Durante a análise das imagens classificadas incorretamente, observamos padrões semelhantes ao Modelo I.

3.3 Modelo III

Inicialmente, aplicamos rotação aleatória nas imagens, com um máximo de 5 graus, adicionando inclinação às imagens para tornar o modelo mais robusto a sinais de trânsito ligeiramente inclinados. Além disso, realizamos translação nas imagens em ambas as direções, com até 10% de deslocamento, para simular a variação na posição dos sinais de trânsito na imagem.

Para tornar o modelo mais resiliente a diferentes condições de iluminação e saturação, ajustamos aleatoriamente a saturação, brilho, contraste e matiz das imagens por diferentes percentagens, simulando variações nos sensores da câmara.

Após o treino, o Modelo III alcançou uma precisão de 98.00%, indicando uma melhoria significativa em comparação com os modelos anteriores. Esta melhoria na precisão sugere que a introdução dessas técnicas de aumento de dados foi eficaz em melhorar a capacidade do modelo de generalizar para novos dados.

Durante a análise das imagens classificadas incorretamente, observamos, no geral, uma redução na ocorrência de problemas relacionados à inclinação e translação. O mesmo não pode ser dito para sinais descoloridos e variações na iluminação das imagens. Isto tudo sugere que as técnicas de data augmentation implementadas foram eficazes em tornar o modelo mais robusto a essas variações mas ainda estão longe de ser ideais.

3.4 Modelo IV

No Modelo IV, realizamos ajustes nas técnicas de aumento de dados implementadas anteriormente, com o objetivo de otimizar ainda mais o desempenho do modelo.

Em particular, reduzimos a alteração na matiz das imagens, enquanto aumentamos a alteração no brilho e no contraste. A redução na alteração da matiz visa minimizar possíveis distorções de cores que possam ocorrer durante o treino. Por outro lado, o aumento no brilho e no contraste visa enfatizar essas características nas imagens, proporcionando ao modelo informações adicionais para distinguir entre diferentes sinais de trânsito com pouquíssima brilho e brilho extremo.

Após o treino, o Modelo IV alcançou uma precisão de 97.98%, uma melhoria marginal em relação ao Modelo III. Embora a diferença na precisão seja pequena, as alterações nas técnicas de aumento de dados ajudaram a melhorar ainda mais a capacidade do modelo de generalizar para novos dados.

3.5 Modelo V

Para o Modelo V, adicionamos técnicas adicionais de data augmentation como :

1. **Transformação de Cisalhamento (Shear Transformation)**: A transformação de cisalhamento envolve o deslocamento de uma parte da imagem em uma direção fixa, mantendo a outra parte estacionária. Esta operação ajuda a introduzir distorções espaciais às imagens, imitando mudanças de perspetiva que podem ocorrer em cenários do mundo real. Ao aplicar a transformação de cisalhamento, nosso objetivo é melhorar a capacidade do modelo em reconhecer sinais de trânsito de diferentes ângulos de visualização.

2. **Ruído Salt and Pepper (Salt and Pepper Noise)**: O ruído salt and pepper é uma forma de ruído aleatório que adiciona pontos pretos e brancos à imagem. Esta técnica de simulação de ruído é comummente usada para imitar ruído de sensor ou imperfeições em dispositivos de aquisição de imagem. Ao introduzir ruído salt and pepper, visamos tornar o modelo mais robusto a artefactos de imagem e melhorar seu desempenho sob condições ruidosas e/ou fraca visibilidade.

3. **Simulação de Bloco Aleatório (Random Block Simulation)**: Esta técnica de aumento envolve a adição de um bloco aleatório à imagem, simulando ocorrências de oclusões ou obstruções que podem ocorrer em cenários práticos. Ao introduzir blocos aleatórios, como danos simulados no poste do sinal ou partículas de poeira bloqueando a lente da câmara, nosso objetivo é treinar o modelo para reconhecer e classificar sinais de trânsito mesmo na presença de oclusões parciais ou obstruções.

Após o treino, o Modelo V obteve uma precisão de 86.42% no conjunto de teste observando uma queda significativa na precisão em comparação com os modelos anteriores. Acreditamos que esta diminuição na precisão pode ser atribuída a uma má escolha de parâmetros nas técnicas de aumento de dados adicionadas.

3.6 Modelo VI

Para o Modelo VI, focamos em otimizar as técnicas implementadas no Modelo V por meio de ajustes manuais nos parâmetros. Com esta abordagem, buscamos melhorar o desempenho do modelo.

Após uma análise cuidadosa, refinamos os parâmetros das técnicas, como a magnitude da transformação de cisalhamento, a intensidade do ruído salt and pepper e o tamanho do bloco aleatório adicionado. Ajustamos esses parâmetros com base em experimentação e avaliação iterativa para encontrar uma combinação que melhorasse o desempenho do modelo.

Após o treino, o Modelo VI alcançou uma precisão de 97.94%, representando uma melhoria significativa em relação ao Modelo V. A otimização manual dos parâmetros das técnicas de aumento de dados permitiu que o modelo capturasse com mais precisão as características relevantes das imagens de sinais de trânsito, resultando em uma melhor capacidade de generalização.

3.7 Modelo VII

Para o Modelo VII, implementamos uma estratégia para aumentar o conjunto de dados concatenando os dados transformados com o conjunto de dados original. Esta abordagem teve como objetivo aumentar o número de pontos de dados disponíveis para treino, fornecendo assim ao modelo exemplos mais diversos para aprender.

No conjunto de dados transformados, aprimoramos ainda mais a variabilidade dos ajustes de brilho e contraste.

Após o treino, o Modelo VII alcançou uma precisão de 98.89%, demonstrando uma melhoria significativa em relação aos modelos anteriores.

Durante a análise das imagens classificadas incorretamente, observamos que muitas delas apresentavam um alto nível de desfoque (blur).

3.8 Modelo VIII

Para o Modelo VIII, adicionamos imagens com efeito de desfoque ao conjunto de dados como parte da estratégia de aumento. Essas imagens apresentam uma aparência suavizada, simulando variações na nitidez das imagens originais.

A inclusão de imagens com efeito de desfoque visa aumentar a capacidade do modelo de reconhecer e generalizar padrões mesmo em imagens com diferentes níveis de nitidez.

Após o treino, o Modelo VIII alcançou uma precisão de 98.99%, indicando uma melhoria adicional em relação aos modelos anteriores. A introdução de imagens com efeito de desfoque contribuiu para fortalecer a capacidade do modelo de generalizar para uma variedade de condições de imagem.

3.9 Modelo IX

Neste modelo, abordamos o problema de classes fortemente desbalanceadas no conjunto de dados. O desbalanceamento de classes pode levar a um treino de modelo enviesado, no qual o modelo pode favorecer a previsão das classes maioritárias em detrimento das classes minoritárias.

Para mitigar este problema, calculamos pesos de classe com base na frequência de cada classe no conjunto de dados. Os pesos de classe são inversamente proporcionais à frequência de cada classe, de modo que as classes menos frequentes recebem pesos mais altos.

Ao atribuir pesos mais altos às classes menos frequentes, aumentamos efetivamente sua influência durante o treino, garantindo que o modelo preste mais atenção a essas classes e aprenda a predizê-las com mais precisão.

Após o treino, o Modelo IX alcançou uma precisão de 98.74%. A precisão ligeiramente inferior do Modelo IX em comparação com o Modelo VIII pode ser explicada pelo trade-off entre as classes maioritárias e minoritárias. O uso de pesos de classe aumentou a atenção do modelo às classes menos frequentes, melhorando sua precisão, mas resultando em uma leve diminuição na precisão das classes maioritárias, afetando a precisão geral. Além disso, a complexidade adicional no treino e a variabilidade na amostragem dos dados podem ter contribuído para esta diferença. Embora a precisão global tenha diminuído ligeiramente, o principal objetivo de abordar o desequilíbrio de classes foi alcançado, melhorando a previsão das classes minoritárias.

3.10 Modelo X

No Modelo X, introduzimos novas técnicas de aumento de dados e ajustamos todas as técnicas usadas anteriormente para melhorar a robustez do modelo em diferentes ambientes.

1. **Barras Verticais e Horizontais:** Adicionamos barras verticais e horizontais geradas aleatoriamente a algumas imagens para simular postes, trilhos ou outras estruturas que possam aparecer na cena. Isso ajuda a diversificar os fundos e o contexto das imagens, tornando o modelo mais robusto a diferentes ambientes.

2. **Ajuste de Técnicas Anteriores:** Refinamos todas as técnicas de aumento de dados aplicadas em iterações anteriores. Isto inclui:

   - Rotação
   - Translação
   - Transformações de cisalhamento
   - Ruído salt and pepper
   - Obstáculo de bloco aleatório

3. **Aumento de Dados Aleatório:** Para cada imagem, geramos três versões adicionais, aplicando de uma a duas das técnicas mencionadas acima de forma aleatória. Esta abordagem cria uma maior diversidade de exemplos no conjunto de treino, ajudando o modelo a generalizar melhor para novas imagens.

Após o treino, o Modelo X alcançou uma precisão de 99.26%. A combinação de técnicas de aumento de dados diversificadas e a geração aleatória de versões aumentadas de cada imagem proporcionaram um conjunto de treino mais robusto e variado, resultando em um desempenho significativamente melhorado.

4. Ensemble

O ensemble é uma técnica que combina as previsões de vários modelos individuais para produzir uma previsão final mais robusta e precisa. No contexto deste projeto, o ensemble foi construído combinando os resultados de três modelos diferentes treinados com abordagens diversas de pré-processamento de dados.

Uma das vantagens principais do ensemble é sua capacidade de reduzir o *bias* e a variância, resultando em previsões mais confiáveis e consistentes. No caso específico deste ensemble, observamos que a maioria das previsões foi feita de forma unânime por todos os modelos, indicando uma alta concordância entre eles. Isto sugere que o ensemble foi capaz de capturar diferentes aspetos e nuances dos dados.

Além disso, o ensemble demonstrou uma excelente capacidade de generalização, alcançando uma precisão impressionante de 99.41% no *dataset* de teste. Mesmo diante de casos onde os modelos individuais falharam, o ensemble ainda conseguiu fazer previsões corretas na maioria dos casos, utilizando o voto da maioria como estratégia de decisão.

No entanto, é importante notar que o desempenho do ensemble depende significativamente da diversidade e qualidade dos modelos individuais que o compõem. Neste projeto, apesar do resultado satisfatório, ficamos aquém dos resultados que seriam possíveis com uma variedade ainda maior de modelos.

5. Conclusão

Este relatório investigou o uso de modelos de Deep Learning aplicados ao conjunto de dados GTSRB (sinais de trânsito alemães) com o objetivo de maximizar a precisão no conjunto de teste.

Na primeira parte do trabalho, foram treinados vários modelos utilizando diversas técnicas de aumento de dados (data augmentation), tanto no pré-processamento quanto em tempo real (dinâmico). Exploramos filtros e métodos de processamento de imagem, avaliando seu impacto no desempenho final da rede. A análise dos resultados permitiu compreender a influência dessas técnicas no desempenho dos modelos.

Na segunda parte do estudo, investigamos o potencial de utilizar ensembles de redes neuronais. Combinamos os modelos treinados na primeira fase através da concatenação de suas saídas. Esta abordagem de ensemble visa aproveitar as previsões individuais dos modelos e combiná-las para obter uma previsão final possivelmente mais robusta.

Em conclusão, este estudo destacou a importância da exploração de técnicas de processamento de imagem para melhorar o desempenho de modelos de Deep Learning no contexto da classificação de sinais de trânsito. Os resultados obtidos fornecem insights valiosos para futuros trabalhos nesta área, demonstrando que abordagens cuidadosas de pré-processamento e combinação de modelos podem levar a melhorias significativas na precisão e robustez dos sistemas de reconhecimento de sinais de trânsito.