Relatório Final - Desafio Pix Force

Marrone Silvério Melo Dantas

No Institute Given

1 Introdução

Este relatório apresenta o desenvolvimento e pesquisa em relação ao problema proposta pela empresa Pix Force como etapa de seleção para a vaga de Desenvolvimento com Visão Computacional. O desafio consiste em desenvolver uma solução que seja capaz de identificar a partir de uma imagem, se ela possui um pinheiro, ou caso contrário possui uma imagem de solo.

Parte da tarefa está em desenvolver uma aplicação/algoritmo que possa identificar essas imagens, obtidas através de drones, de forma automática e retornar a identificação dessa imagem. As imagens originais possuem grandes dimensões, para um processamento mais rápido ela forma divididas em mosaicos, com dimensões 50x50. Das entregas esperadas estão:

- Analise exploratória dos dados: contida neste documento e ***data_exploration.ipynb*** no repositório do desafio;
- Aplicação: arquivos que estão presentes no repositório do desafio, assim com um arquivo guiam para execução ***HOWTO.MD***;
- Relatório: dados contidos dentro deste arquivo.

As seções a seguir cobrirão os conteúdos das entregas restantes.

2 Exploração de Dados

Primeiramente precisamos selecionar o tipo de problema e abordagem que iremos seguir a partir dos dados. Em nosso caso temos um declarado problema de classificação. Possuimos como uma entrada e desejamos saber a qual classe essa imagem pertence. As possíveis classes são "soil" e "tree", a classe "tree", são aquelas que possuem um pinheiro, as restantes são classificadas como "soil". A Figura 1, demonstra alguns exemplos de itens presentes na base de dados.

Fig. 1: Exemplo de imagens na base de dados. Acima temos exemplos de imagens "soil", e abaixo temos exemplos de imagens "tree".

Como podemos observar na Figura 1, podemos executar algumas avaliações qualitativas com relação as imagens. A maioria das imagens que possuem árvores estão centralizadas, e possuem uma forma aproximadamente circular. Além da maioria das árvores estarem presentes com uma certa distância em relação a outras, talvez sendo um indício de uma distância mínima de plantação, o que pode ser um atributo adicional em um processo de treinamento. As imagens classificadas com "soil", em geral podem possuir uma região "verde", que contém algum tipo de gramado, ou até mesmo um porção parcial de uma árvore. Por possuir essa informação mista, talvez uma solução baseada em segmentação de cores seja bastante limitada. Outro ponto que pode se perceber está contido nas imagens de canto. Nesse tipo de imagem ao se efetuar o corte, para geração dos azulejos, em alguns casos se apresenta um tipo de borda preta, que não condiz com a condição rela do ambiente, sendo algo que pode atrapalhar no treinamento, já que é uma dado que não representa a distribuição original.

Tendo isso em mente, outro ponto a ser considerado são os casos com marcação errada, ou caso complexos. Defino como casos complexos, aqueles casos, que para uma pessoa não treinada, não é possível distinguir qual a classe do azulejo, e os com marcação erradas aqueles que estão com as anotações pertencentes a outra classe. Esse ponto é delicado, sem um conhecimento prévio não é possível afirmar com certeza, se as imagens estão marcadas erroneamente ou são somente casos complexos. Na Figura 2 temos alguns exemplos desses casos.

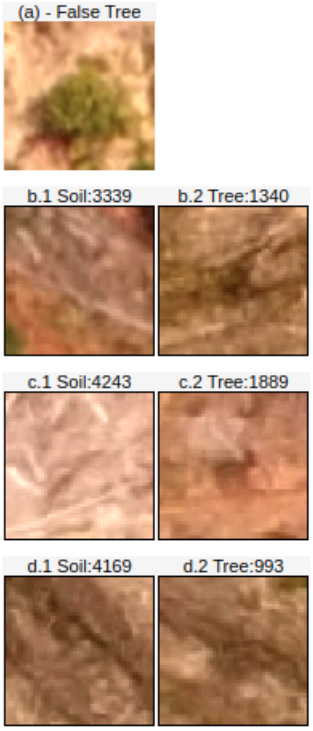


Fig. 2: Exemplo de casos complexos e/ou marcados erroneamente.

Como podemos observar na Figura 2, temos alguns casos complexos. Em primeiro momento temos a imagem (a), que possui uma grande vegetação, e possui um certo formato circular, porém é classificada como "soil". Em segundo lugar, temos os casos b, c e d. Estes casos na segunda coluna apresentam imagens que forma classificadas como "tree", porém no melhor dos casos, não é possível para um olho não afirmar que existe uma árvore nesta localização. Sendo algo que também pode dificultar algum tipo de treinamento.

Mesmo com uma distribuição diversificada é notável que as imagens com "tree", possuem um tom de verde mais proeminente, como avaliação complementar, foi adicionar uma avaliação de histograma, para determinar a influência das cores na definição em cada imagem. Como as imagens são coloridas, ou seja, estão dentro do padrão RGB, não é possível gerar um histograma para esse tipo de imagem. Duas saídas são: gerar o histograma a partir de uma versão em escala de cinzas, ou gerar um histograma para cada canal de cor (verde, vermelho e azul). Como cada uma possui as suas nanicas, foi decidido avaliar ambas. Para a geração da métrica, for gerado o histograma de todos os canais, e da versão em escala de cinza, e por fim foi retirada a média de todos os histogramas por classe. Figura 3 demonstra os gráficos referentes aos histogramas descritos.

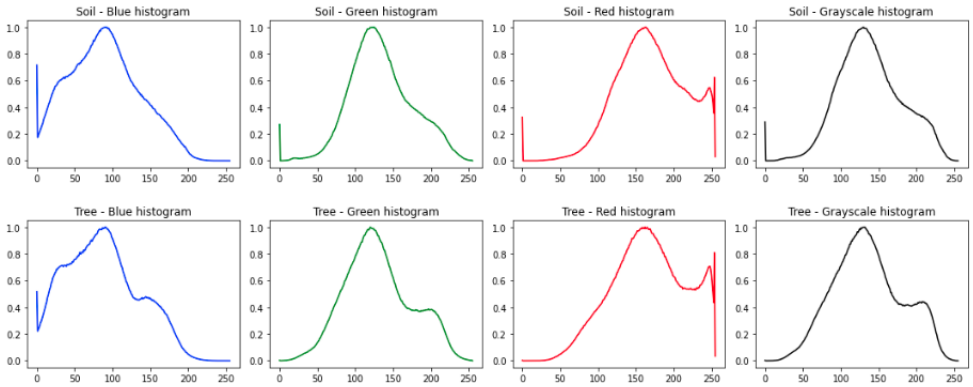


Fig. 3: Histogramas por cor e escalada de cinza. Da esquerda para direita azul, verde, vermelho e escala de cinza.

Na Figura 3, a linha superior contem os histogramas para a classe "soil" e linha inferior para a classe tree. Como podemos observar os histogramas apresentam uma morfologia muito similar, entre ambas classes, sendo difícil definir uma classe somente pelo indício do histograma. Mesmo com um outro sistema de cores com HSV, não seria trivial determinar um trehshold manual que tornasse viável qualquer tipo de inferência. Porém, dessa avaliação foi adquirido um dado importante, a classe "soil", possui alguns picos perto do zero, nos histogramas da cor "verde" e "vermelho". Esses picos apesar de não serem decisivos, podem ser utilizados como características extras para uma classificação.

Por fim temos uma avaliação quantitativa, a base de dados possui uma desbalanceamento forte, com a classe "soil" contendo o dobro de exemplos da classe "tree". A classe soil possui 4448 exemplos, enquanto a classe "tree", possui 2224. Como a classe é desbalanceada, pode ocorre uma viés em algum possível treinamento. Uma possível solução é a utilização de geração de dados sinteticos (data augmentation), ou alguma técnica de redução de dados, como random undersampling.

Como conclusão da avaliação temos:

- As árvores possuem forma aproximadamente circulares, um atributo que pode ser adicionada;
- Um limpeza de dados poderia ser benéfica, mas sem o conhecimento prévio, exemplos que são somente complexos podem ser eliminados;
- Os histogramas foram inconclusivos, mas apresentaram mais um possível atributo para o treino;
- A centralização e distância das árvores podem ser utilizada como atributo extra;
- A base é desbalanceada, deve ser usada alguma técnica de geração ou redução de dados.

3 Proposta

Foi decidido como primeira abordagem recorrer a uma solução bastante utilizada, aplicação de transfer learning com um modelo conhecido. O modelo selecionado foi o VGG16, que ainda se apresenta como um arquitetura viável. Nesse passo somente as imagens foram utilizadas como entrada para rede. Os pesos utilizados foram os disponibilizados pelo Keras, treinados na ImageNet. As imagens foram mantidas na mesma dimensão de 50x50. Para o treinamento foi criado um processo de data-augmentation, que foi executado durante o treinamento.

A definição de data augmentation consiste em gerar novas imagens a partir de antigas, de forma que ainda representem de alguma forma a distribuição original. Tendo isso em mente, foi limitado o total de transformações possíveis, por exemplo, um data augmentation possível seria cortar uma imagem e deslocar do centro, algo que não é tão presente na base original. Por isso nos limitamos as seguintes transformações:

- Rotação a partir do centro, podendo inclusive simular os cortes de imagens de cantos da base;
- Rotação horizontal;
- Rotação vertical;
- Mudança de brilho (aumento e diminuição)

A geração de novas imagens pode contribuir para uma maior generalização do modelo. Com essa técnica podemos simular ambiente que não estão presentes em um conjunto, mas representam parte de outro. A Figura 4 apresenta dois exemplos de imagens com data augmentation.

Fig. 4: Exemplo de imagens geradas por data augmentation.

Na abordagem com transfer learning, foram mantidas as camadas convolucionais como ferramenta de extração de características, e a rede foi retreinada, adicionando

novas camadas totalmente conectadas. Para o treinamento a base de dados foi dividida em três conjuntos, de forma estratificada: treino, teste e validação, divididos da seguinte forma:

- Treino: 80% da base com 5337 amostras, sendo 1782 de "tree" e 3555 de "soil";
- Test e Validação: Cada uma contendo 10% da base com 668 amostras, sendo 220 de "tree" e 447 de "soil";

O modelo VGG16 foi treinado por 60 épocas, onde somente os melhores resultados na validação forma salvos. Foi adicionada uma flag para redução da taxa de aprendizagem, caso a rede parasse de "aprender". Na última camada a função de ativação selecionada foi a sigmoid, com a função de erro de *binary_crossentropy*. Infelizmente os resultados não foram muito promissores, foi alcançada uma acurácia média de 87.67%. O que pode ser um indício de que a base mesmo com data augmentation não foi suficiente para os parâmetros da VGG16. Outra possibilidade seria o retreino de todos os parâmetros da rede, algo que poderia tomar muito tempo. Com esse resultado em mão decidimos seguir para uma abordagem paralela, em que foi reduzido o total de parâmetros.

3.1 VGG-Based

A VGG-Bases, foi uma rede gerada como proposta para esse problema, ela é uma versão simplificada da VGG16, ela possui somente 4 camadas convolucionais, com uma camada de max-pooling após cada convolução e finalmente 4 camadas totalmente conectadas. A Figura 5 demonstra como a arquitetura foi projetada.

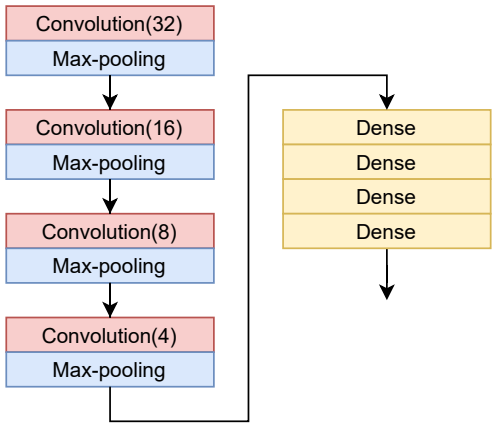


Fig. 5: Exemplo de imagens geradas por data augmentation.

Como podemos ver na Figura 5, a rede foi bastante reduzida, saindo da casa dos milhões de parâmetros para somente 233469. Seguimos o mesmo método de treinamento, com aplicação de data augmentation, loss, função de ativação e divisão de bases.

Com esse modelo obtivemos resultados mais expressivos, chegando a uma acurácia de 92.03%, um acréscimo de 4.36 pontos percentuais. Esse modelo demonstrou o quanto a redução de parâmetros foi benéfica, e que as características estão sendo extraídas de forma mais concisa.

3.2 Mixed-model

O modelo anterior mixed-model, apresentou um bom desempenho. Porém guiados pela nossa abordagem inicial, podemos observar que os histogramas possuíam um poder de descrição, mesmo que baixo, da base de dados. Então foi proposta a união dessa característica com a nossa rede. A nova rede nomeada mixed-model, contém a rede anterior vgg-based em união com a soma dos histogramas como nova característica. Figura 6 demonstra a arquitetura do modelo.

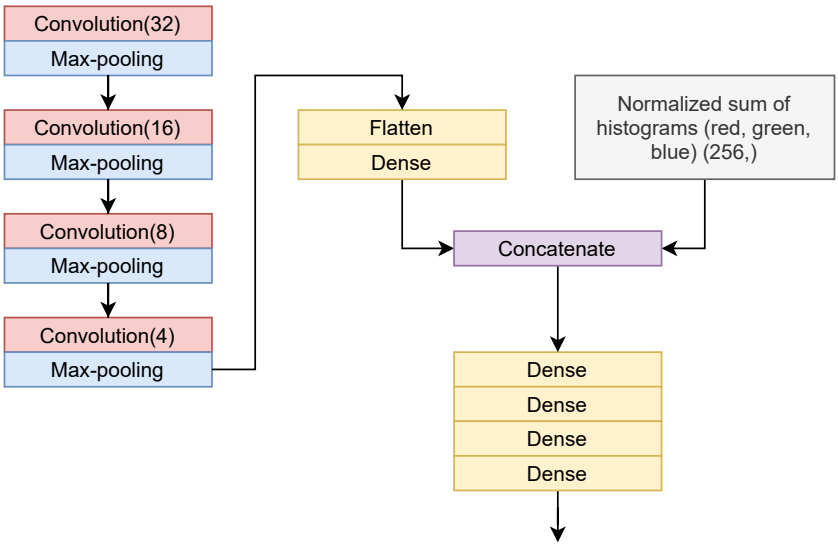


Fig. 6: Arquitetura mixed-model.

Como podemos observar na mixed-model, temos a rede base a esquerda, em que durante o nosso treino funcionou como extrator de características, em seguida uma camada flatten para planificação dos dados, e uma camada densa para redução dos dados. Nesse momento é que ocorre uma concatenação, da saida densa da rede, com a soma histogramas gerados. De cada imagem foi extraída o conjunto de histogramas para cada canal, então seus valores forma somados por valor de intensidade, resultando em um array com 256 posições. Esse array do histograma foi normalizado para inserção na rede. Seguimos o mesmo protocolo de treinamento das redes anteriores, com a mesma di-

visão de dados. Felizmente com esse novo modelo, obtivemos um melhor desempenho, chegando a uma acurácia de 93.68%, um ganho de 1.65 pontos percentuais.

4 Análise dos Resultados

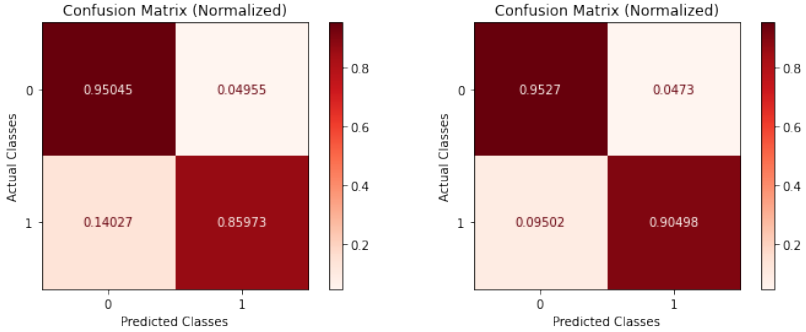
Nos experimentos foram dois modelos base, um modelo nomeado vgg-based, que alcançou 92.03% em acurácia, e o mixed-model, que alcançou 93.68%. Apesar de uma parente pequena diferença entre os resultados dos modelos, é válido lembrar que a nossa base é desbalanceada, então um resultado enviesado pode gerar uma acurácia maior, guiando a uma interpretação errada dos dados.

Como ferramenta auxiliar avaliamos duas medidas complementares, taxa de verdadeiros positivos e área sobre a curva roc. A taxa de verdadeiro positivos (TPR) indica o quão o modelo está correto em indicar que a amostra pertence aquela classe, enquanto a curca roc indica o o pode de separabilidade do modelo, ou seja, o quão ele consegue distinguir ambas as classes. Tabela 1 demonstra os resultados obtidos.

Table 1: Comparação medidas complementares entre Mixed-Model x VGG-Based

Model	TPR - Soil	TPR - Tree	AUC ROC
Mixed-Model	**0.952**	**0.904**	**0.928**
VGG-Based	0.950	0.859	0.905

Como podemos observar, a rede VGG-Based realmente possuía um pequeno viés para a classe "oil", o que pode ter influenciado os resultados da acurácia. O modelo Mixed-Model aparenta ter reduzido esse viés, onde é possível observar na taxa de true positives da classe "tree", obteve um incremento de mais de cinco pontos, provavelmente causado peal adição da descrição extra dos histogramas no treinamento. Foram plotadas as matrizes de confusão, onde essas evidências também se apresentam claramente. A Figura 7 demonstra as matrizes de confusão.



(a) VGG-Bases (b) Mixed-Model

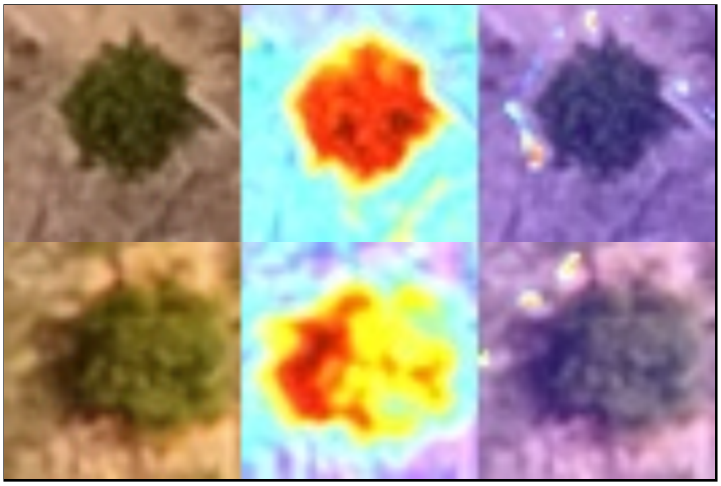
Fig. 7: Matrizes de confusão. Em (a) temos a matriz de confusão do modelo VGG-Base, em (b) temos a matriz para o modelo Mixed-Model.

Como podemos observar, o mixed-model conseguiu reduzir o viés, já que o modelo diminuiu a frequência em que executava predição como sendo da classe "soil", além de incrementar a taxa de acerto em ambas as classes.

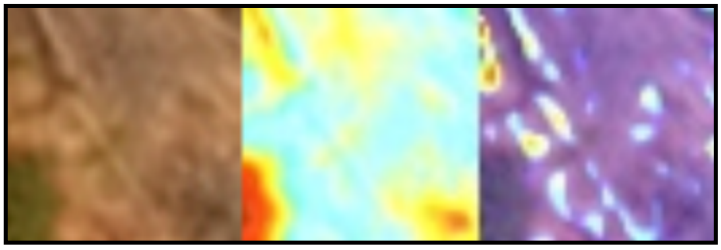
4.1 Explainable AI

Um passo por vezes essencial, se relaciona a entender o que a rede aprendeu, e para onde ela está olhando no decorrer da tomada de decisão. Para entender essa informação existem diversas abordagens, a que foi utilizada é denominada mapa de ativação, que utiliza o logist da última camada convolucional, para determinar a importância de cada feature. Foi utilizada uma implementação baseada na implementação de Nick Biso[1], onde alguns detalhes foram modificados para ser compatível com uma rede mista, com múltiplas entradas. Figura 8 demonstra alguns exemplos de ativações.

[1] Disponível em https://github.com/nickbiso/Keras-Class-Activation-Map



(a)



(b)

Fig. 8: Mapa de ativações. Em (a), exemplos com classe "tree", em (b) exemplo com classe "soil".

Como podemos observar o modelo proposto realmente está tomando a decisão baseada na região desejada, em (a), temos exemplos com alta ativação nas regiões onde as árvores se encontram, em (b) que temos um exemplo com "soil", temo poucas ativações.