

Classificação de Imagens Noite-Dia

Camila Moura
Jonlenes Castro
Prof.: Anderson Rocha

Resumo

Este trabalho propõe uma solução ao problema de classificação de imagens em dia ou noite. Inicialmente desenvolvendo, com base em quatro datasets distintos, um novo dataset em que as rotulações aquedadas para auxiliar na resolução do problema foram inseridas. A partir deste novo dataset foram construídos modelos utilizando a associação de classificadores, distintos métodos de extração de features, transfer learning sobre diferentes CNNs treinadas em diferentes datasets e meta learning. O modelo final resultante dessas combinações obteve acurácia de aproximadamente 98%.

1. Introdução

Utilizar máquinas para resolver problemas e tomar decisões, sem que haja o constante monitoramento humano, é parte do desafio que impulsiona o desenvolvimento de pesquisas em aprendizagem de máquina. Para tanto, algumas técnicas e modelos foram desenvolvidos, de modo a auxiliar esta aprendizagem.

Em face disto, algumas destas técnicas de aprendizagem de máquina foram utilizadas neste trabalho, no intuito de resolver o problema de classificação de imagem entre dia e noite, que consiste em identificar, independentemente das condições climáticas, se uma imagem foi capturada durante o dia ou a noite.

A proposta apresentada neste refere-se a realização da aprendizagem de máquina, em que esta é submetida a um *dataset* com diversas imagens e então realiza-se a classificação entre elas com base no período em que foram capturadas.

1.1. Trabalhos relacionados

Alguns trabalhos já foram desenvolvidos visando solucionar problema similar ao deste trabalho, com um enfoque maior em identificar, a partir de diferentes períodos do dia, o mesmo lugar, como apresentado pelo [1].

Entretanto, não foi encontrado um *dataset* rotulado para

realizar a classificação apenas em períodos do dia, sendo necessária a utilização de alguns *datasets*, como especificado na Seção 1.2, para a construção de um outro *dataset*, sendo este utilizado para o desenvolvimento deste trabalho.

1.2. Datasets

Para a realização desse trabalho foram utilizados como *datasets* o *Transient* [2], o AMOS [3], o Barcelona [4] e o *Places* [5].

O primeiro *dataset* a ser utilizado para este problema é o *Transient*, pois contém 8571 imagens de 101 *webcams*, já rotuladas com 40 *labels*.

Em seguida, utiliza-se um subconjunto do AMOS, tendo em vista que em sua totalidade ele contém 1.128.087.180 imagens, extraídas por 29945 *webcams* rotuladas com o dia e hora da captura. Já o *Places* contém mais de 10 milhões de imagens, incluindo cerca de 400 categorias de cenas e o Barcelona, contém apenas imagens capturadas na cidade de Barcelona com boa resolução.

2. Construção do dataset

Conforme mencionado por [2], os atributos do *dataset Transient* possuem seus valores variando entre 0 e 1, em que valores menores que 0.2 são considerados fortes negativos e valores maiores que 0.8 são considerados fortes positivos. Embasando-se nesta afirmação e nos experimentos realizados, definiu-se o critério inicial para classificar os exemplos deste *dataset* em dia ou noite, sendo utilizados como atributos o *daylight*, que representa a quantidade de luz do dia contida na imagem e o *night* que consiste na representação da noite.

Com isso foi possível classificar as imagens entre dia e noite, a partir do percentual de cada atributo, distribuídos da seguinte maneira: *daylight* > 0.8 e *night* < 0.2 atribui-se a classe dia; *daylight* < 0.2 e *night* > 0.8 atribui-se a classe noite.

A partir desta configuração foi realizada a classificação de 40% do *Transient*, entretanto, dentre os exemplos que foram classificados como noite (523), verificou-se que uma parcela significativa deles representavam momentos de

transição (fim de tarde/início do dia e nascer/pôr do sol), onde, até mesmo pela identificação manual era difícil realizar a classificação da imagem em dia ou noite.

Com o intuito de solucionar este problema, outra classe foi adicionada ao processo de classificação desse trabalho, intitulada transição, que representa justamente os momentos anteriormente mencionados. Justaposto a isso foram acrescidos alguns atributos para auxiliar a realização da classificação do *Transient*, sendo *sunrisesunset* (nascer e pôr do sol), *dawndusk* (amanhecer-anoitecer) e *sunny* (ensolarado).

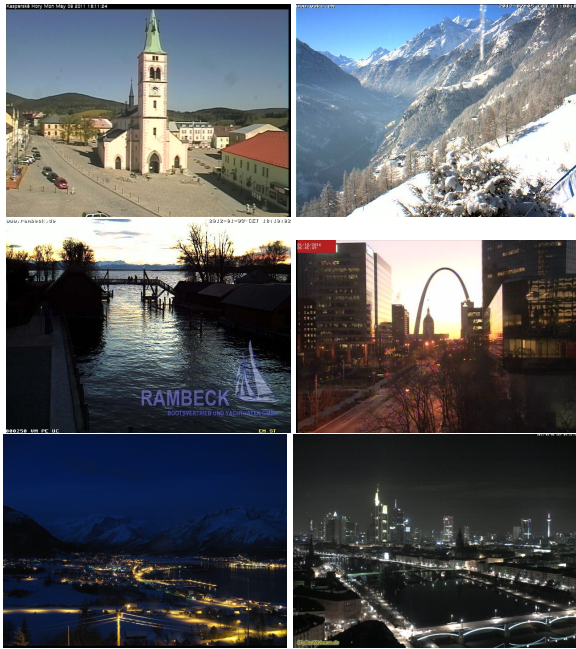


Figura 1. Exemplos de cada classe

Considerando as 3 classes e após alguns experimentos, uma nova configuração para a classificação do *Transient* foi definida, como mostradas a seguir:

- Extração das imagens com *sunrisesunset* > 0.8 ou *dawndusk* > 0.8 , para representar os momentos de transição (classe *Transition*);
- Após a remoção das imagens de transição, realizar a extração das imagens com *night* > 0.8 , que representam as imagens noturnas (classe *Night*);
- Nas imagens restantes, extrai-se aquelas com *daylight* > 0.8 ou *sunny* > 0.8 , que representam as imagens de dia (Classe *Day*).

Após este experimento, foi possível classificar 660 imagens como transição, 396 como noite e 4195 como dia, o

que representa apenas 65% do *Transient*, porém, ao ser realizada a conferência manual das classes, foi verificado que apesar da divisão ter melhorado em relação a anterior, ainda continha várias classes incorretas.

Ao ter sido realizado alguns testes com os atributos, decidiu-se selecionar todos os exemplos corretos da classe noite, no total de 332 exemplos e selecionar nas outras duas classes apenas 332 exemplos também, que serão utilizados para o treinamento dos classificadores, para rotular novas imagens e assim aumentar o *dataset*.

3. Rotulação de novas imagens

Como esse *dataset* rotulado, foi realizada a extração de *features* e treinamento de alguns classificadores para ajudar no processo de rotulação e aumentar o *dataset* atual. Ressalta-se que antes de realizar os treinamentos mencionados nesta seção, foi realizado o processo de *feature scaling* [6], assim como foram utilizados parâmetros de regularização para cada classificador. Já a extração de *features* foram utilizadas imagens redimensionadas em 512 x 512 pixels.

3.1. Features extraction

Para dar início aos experimentos foram utilizados os métodos de cálculo do histograma e *gray-level co-occurrence matrix* (GLCM) da imagem, sendo o histograma [7] calculado para cada canal de cor da imagem (RGB), produzindo 256 *features* por canal, totalizando 768 *features*.

Para a extração de *features* utilizando o GLCM [8], a imagem teve sua quantidade de tons reduzidos de 256 para 32 a fim de diminuir a quantidade de *features*, e ao aplicar o GLCM nessa nova imagem foi extraída uma matriz de 32x32x3 que gerou 3072 *features*.

Os resultados dos experimentos podem ser vistos na Tabela 1, onde (I) representa o histograma e (II) o GLCM. A métrica apresentada na tabela é o f1 score [9] sobre 20% dos dados divididos utilizando *cross validation* [10].

Tabela 1. Resultados da classificação

Classificador	I	II	I + II
Linear SVM	84.81%	88.70%	90.85%
Decision Tree	82.17%	82.52%	85.39%
Random Forest	88.89%	88.87%	90.81%
Neural Network	87.72%	92.18%	91.74%
Naive Bayes	79.05%	68.89%	69.66%
Logistic Reg	84.80%	86.66%	90.69%

Em seguida, foram extraídos *features* utilizando o *Local Binary Pattern* (LBP) [11] com 100 pontos e raio 25. Do resultado do LBP foi computado um histograma normalizado com 100 *bins*, que foram utilizados como *features*. Este processo é repetido para cada canal da imagem o que totaliza

300 valores. Os resultados dos experimentos podem ser vistos na Tabela 2, onde (III) representa a utilização dos *features* extraídos com LBP.

Tabela 2. Resultados da classificação

Classificador	III	II + III	I + II + III
Linear SVM	82.82%	89.39%	90.15%
Decision Tree	76.66%	84.44%	84.67%
Random Forest	80.00%	91.36%	90.72%
Neural Network	84.92%	92.89%	92.28%
Logistic Reg	83.18%	87.98%	88.57%

3.2. Features from deep model

Para os experimentos de extração de *features* utilizando *deep models*, foram utilizados o ResNet50, InceptionV3 (IncV3) e Xception (Xcp) pré-treinados com o *dataset* ImageNet.

Em cada um deles foi removida a camada *fully-connected* (camada do topo) e os valores antes repassados para esta camada foram utilizados como *features* para treinar os classificadores. A Tabela 3 apresenta os resultados (f1 score) utilizando estes modelos.

Tabela 3. Classificação - Feature extration com CNN

Classificador	ResNet50	IncV3	Xcp
Linear SVM	94.77%	92.66%	91.85%
Random Forest	91.87%	79.39%	76.17%
Neural Network	95.73%	91.68%	90.20%
Logistic Reg	95.24%	68.79%	80.07%

3.3. Incrementando o dataset

Como pode ser visualizado na subseção anterior, determinados conjuntos de *features* apresentam melhores resultados dependendo do classificador utilizado. Em face disto e considerando que a rotulação de dados é crucial e não deve conter erros, serão utilizados vários classificadores com diferentes *features*.

Para esta etapa, foram selecionados os classificadores e seus respectivos *features*, que obtiveram melhores *f1 score*, sendo todos treinados e utilizados na rotulação. O modelo representativo deste processo pode ser visualizado na Figura 2, que contém o classificador e seus *features*.

De acordo com o referido modelo, uma nova imagem será adicionada ao *dataset* se, além de todos os classificadores rotulá-la como sendo da mesma classe, a rotulação manual também for positiva.

Esse processo é executado recursivamente de modo a realizar o treinamento dos classificadores com o *dataset* atual e calcular o *f1 score* sobre os 20% de dados da validação, a fim de certificar que os classificadores estão com boa precisão. Em seguida, adiciona-se as novas imagens e repete-se todo o processo novamente até que não haja mais imagens

para serem rotuladas ou até que se obtenha um *dataset* com tamanho aceitável.

Devido o processo de rotulação do *dataset* ser trabalhoso e demandar muito tempo, foi definido um limite de dados para compor o *dataset*, sendo este de aproximadamente 9 mil imagens, ou seja, 3 mil de cada classe. As imagens utilizadas para compor este *dataset* foram extraídas do *Transient*, do AMOS e do Barcelona.

Com o intuito de obter maior diversidade das imagens, tendo em vista as características específicas de cada *dataset*, dado que o *Transient* apresenta imagens em cenários abertos e com diferentes condições climáticas, o AMOS imagens de monitoramento com baixa resolução, e o Barcelona com registros urbanos em cenários menos amplos e com boa resolução, foram combinadas amostras de cada *dataset* para cada classe. Também com esse objetivo, as imagens extraídas do AMOS e do *Transient* foram selecionadas aleatoriamente, pois eles contém imagens bastante parecidas indicativa de sequencialidade.

Após a junção dos *datasets* obteve-se uma nova base de dados com diferentes cenários e condições climáticas, e assim, por estar mais diversificado, apresenta melhores informações para o treino. Para realizar o teste do modelo, foram selecionadas um mil imagens com a mesma proporção por classes.

4. Aprimoramento do modelo

Para a construção do modelo final deste trabalho, inicialmente será retreinado o modelo utilizado na rotulação de dados e acrescida a utilização de *meta learning*.

4.1. Meta Learning

O retreinamento do modelo, representado pela Figura 2, com o *dataset* completo obteve melhores resultado, sendo descritos na Tabela 4 os valores por cada classificador enumerados de 1 a 7.

Tabela 4. Classificação com o dataset completo

Classificador	accuracy	f1-score
1	94.39%	94.39%
2	95.79%	95.76%
3	94.20%	94.20%
4	95.98%	95.96%
5	94.84%	94.80%
6	97.51%	97.50%
7	97.45%	97.44%

Na próxima etapa, os resultados obtidos são repassados a outro classificador que será utilizado para definir a classe final para cada exemplo. Ao ser utilizado o *meta learning* novos e melhores resultados foram obtidos, como pode ser observado na Tabela 5, se comparado com o voto majoritário.

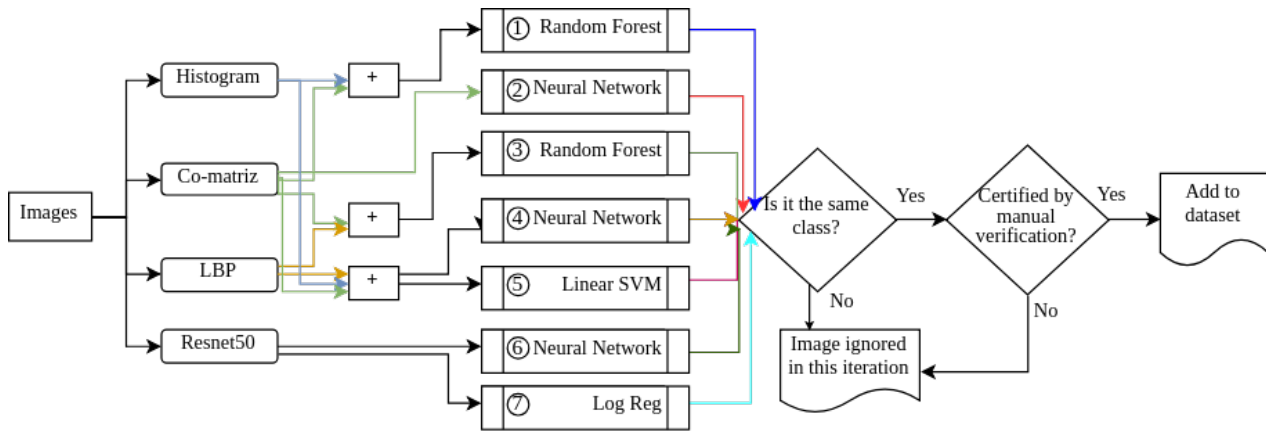


Figura 2. Rede de classificadores

Tabela 5. Meta learning

Classificador	accuracy	f1-score
Linear SVM	97.13%	97.13%
Decision Tree	98.08%	98.08%
Random Forest	98.02%	98.02%
Neural Network	96.68%	96.70%
Logistic Reg	96.81%	96.82%

4.2. Transfer Learning - ResNet50 com ImageNet

Como a extração de *features* pelo ResNet50 treinado com o ImageNet apresentou bons resultados, e tendo o *dataset* com maior quantidade de imagens, prosseguiu-se o experimento para utilizar *transfer learning* por meio do *fine-tuning*, que consiste no processo de ajustar as redes treinadas em um *dataset* para o treinamento com o outro *dataset* selecionado [12].

A *fully connected layer* do ResNet50 foi removida e substituída por novas camadas, conforme apresentado na Figura 3, sendo estas escolhidas devido a apresentarem os melhores resultados, após ser realizados experimentos com outras configurações.

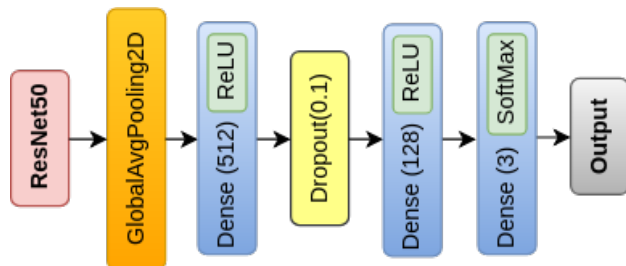


Figura 3. Fully connected layer para ResNet50.

Como resultado desse processo treinado por sete *epochs*, obteve-se 90% de *accuracy* e 0.2789 de *loss* no conjunto de treinamento, e 80% de *accuracy* e *loss* de 1.2179 no

conjunto de validação. Ajustando este modelo, conseguiu-se atingir uma *accuracy* de até 98% sobre o conjunto de treinamento, porém na validação este resultado apresentou decréscimo indicativo de *overfitting* sendo a *accuracy* de apenas 60%.

4.3. Transfer Learning - VGG16 com Places

Para este experimento foi utilizado umas das CNNs treinadas com o *dataset Places* disponibilizadas por [5], sendo denominada VGG16.Places365. Esta CNN consiste em uma VGG16 com *fine-tuning* para as classes dos *Places*, ou seja, 365 classes. A escolha desta CNN justifica-se devido ao conteúdo das imagens serem mais apropriadas para cenário desse trabalho, visto ser imagens para reconhecimento de cenários.

O *fine-tuning* realizado nesta CNN, assim como o anteriormente utilizado, remove as camadas do topo (fully connected layer), e com a saída obtida pelo VGG, adiciona novas camadas, conforme apresentado na Figura 4, para então realizar a classificação.

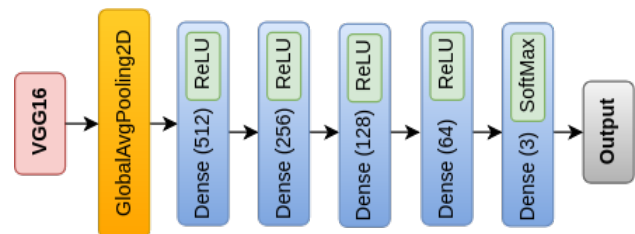


Figura 4. Fully connected layer para VGG.

Após realizar o treinamento por uma *epoch* tendo 90% das camadas do modelo congeladas e utilizando *optimizer rmsprop*, obteve-se os resultados de apresentados na Tabela 6. Ressalta-se que adicionar mais *epochs* não representou melhora significativa dos resultados.

Tabela 6. Resultados com o VGG16.Places365			
Treinamento		Validação	
Loss	Accuracy	Loss	Accuracy
0.3222	97.14%	0.3469	96.07%

4.4. Modelo Final

Para o modelo final, foram utilizados os sete classificadores definidos na Seção 4.1, acrescido com o VGG16.Places365 com *fine-tuning*, que foi escolhido por ter sido treinado com *dataset Places* e por apresentar *accuracy* superior a alguns dos outros classificadores. Neste ponto, descartou-se o ResNet50 com *fine-tuning*, pois além de não ter apresentado bons resultados ele já esta sendo utilizado na extração de *features*.

O resultado obtido com a rede de oito classificadores, é enviado a outro classificador (*Logistic Regression*) no final da rede, apresentando *accuracy* de 98.83% e *f1-score* de 98.53, sobre o conjunto de validação.

5. Resultados

Para o modelo anterior, foi realizada a predição sobre o conjunto de teste (aproximadamente mil imagens). Os resultados obtidos podem ser visualizados na Figura 5 enquanto na Tabela 7 apresenta-se a *confusion matrix*.

	precision	recall	f1-score
Dia	0.98	1.00	0.99
Noite	0.99	0.97	0.98
Transição	0.96	0.96	0.96
avg / total	0.98	0.98	0.98

Figura 5. Report.

Tabela 7. Confusion matrix			
	Predict Dia	Predict Noite	Predict Trans
Dia	589	0	1
Noite	3	562	15
Trans	9	7	384

A fim de realizar a verificação do modelo desenvolvido neste trabalho, efetuou-se o teste sobre o *dataset* de um problema sobre outro domínio, o Nexet [13], sendo este composto por 50 mil imagens de estradas, capturadas a partir do interior de veículos.

Apesar do Nexet ter sido construído para auxiliar na detecção de veículos, algumas rotulações contidas nele, dentre elas o *Day*, *Night* e *Twilight*, sendo essa última parcialmente equivalente a classe de transição deste trabalho,

auxiliaram no desenvolvimento deste. Considerando apenas esses atributos foi realizada a classificação de 10 mil imagens do Nexet, selecionadas aleatoriamente (1/3 para cada classe), obtendo 83% de *accuracy* e 82% de *f1-score*, demonstrando que o modelo conseguiu ser generalizável.

6. Conclusão

Mesmo apresentando bons resultados com o *dataset* desenvolvido, algumas imagens continuaram não sendo corretamente classificadas, como pode ser visto na *Confusion matrix* da Seção 5. Então, com o intuito de identificar a causa destas falhas, realizou-se a análise manualmente das imagens, onde também não foi possível identificar claramente a qual classe estas pertenciam.

Foi identificado que as imagens que apresentaram maior dificuldade em serem corretamente classificadas, geralmente continham excesso de nuvens ou condições climáticas desfavoráveis e, em alguns casos, pertenciam a locais com presença de neve, tornando confusa a identificação de um período do dia. Alguns exemplos em que o treinamento não obteve êxito podem ser conferidos na Figura 6.

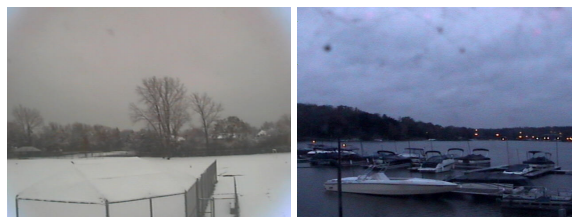


Figura 6. Exemplos de imagens classificadas erroneamente

Um dos problemas encontrados durante o desenvolvimento deste trabalho, não sendo completamente solucionado, consiste na definição de um limite para cada classe, pois como foi definida a divisão em três classes distintas, recorrentemente uma imagem é classificada em uma classe distinta da qual deveria compor, dada a tênue divisão entre as classes. Este problema é especialmente difícil de ser resolvido, dada a quantidade de dependências para se decidir entre uma classe, haja vista que a imagem de um mesmo lugar não é invariante a mudanças climáticas ou aos parâmetros orbitais de Milankovitch [14].

Sugere-se como uma possível melhoria a utilização de um modelo que considere a detecção de sombras, por exemplo, e a partir dela consiga prever a qual período a imagem pertença. Também poderia ser empregado um meio de eliminar as nuvens presentes na imagem de modo a facilitar o processo de classificação.

Referências

- [1] Hao Zhou, Torsten Sattler, and David W Jacobs. Evaluating local features for day-night matching. In *European Conference on Computer Vision*, pages 724–736. Springer, 2016. 1
- [2] Pierre-Yves Laffont, Zhile Ren, Xiaofeng Tao, Chao Qian, and James Hays. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Transactions on Graphics (proceedings of SIGGRAPH)*, 33(4), 2014. 1
- [3] Nathan Jacobs, Nathaniel Roman, and Robert Pless. Consistent temporal variations in many outdoor scenes. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–6. IEEE, 2007. 1
- [4] Joseph Tighe and Svetlana Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. In *European conference on computer vision*, pages 352–365. Springer, 2010. 1
- [5] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 1, 4
- [6] Sebastian Raschka. About feature scaling and normalization. http://sebastianraschka.com/Articles/2014_about_feature_scaling.html, 2014. 2
- [7] R. Fisher, S. Perkins, A. Walker, and E. Wolfart. Intensity histogram. <https://homepages.inf.ed.ac.uk/rbf/HIPR2/histogram.htm>, 2013. 2
- [8] David A Clausi. An analysis of co-occurrence texture statistics as a function of grey level quantization. *Canadian Journal of remote sensing*, 28(1):45–62, 2002. 2
- [9] Renuka Joshi. Precisão, precisão, recall e f1 pontuação: Interpretação de medidas de desempenho. <http://blog.exsilio.com/all/f1-score-interpretation>, 2016. 2
- [10] Sunil Ray. Improve your model performance using cross validation. <http://aiweb.techfak.uni-bielefeld.de/content/bworld-robot-control-software/>, 2015. 2
- [11] Adrian Rosebrock. Local binary patterns with python & opencv. <https://www.pyimagesearch.com/2015/12/07/local-binary-patterns-with-python-opencv/>, 2015. 2
- [12] Felix Yu. A comprehensive guide to fine-tuning deep learning models in keras. <https://flyyufelix.github.io/2016/10/03/fine-tuning-in-keras-part1.html>, 2016. 4
- [13] Nexar. Vehicle detection in the wild using the nexet dataset. <https://www.getnexar.com/challenge-2/upload/>, 2017. 5
- [14] Michelle Simões Reboita, Angelita de Paula Pimenta, and Ulisses Antônio Natividade. Influência da inclinação do eixo de rotação da terra na temperatura do ar global. *Terrae Didactica*, 11(2):67–77, 2015. 5