"A frase mais excitante de se ouvir na ciência, que leva a maioria das descobertas, não é 'Eureka', mas sim 'que esquisito!' " (Isaac Asimov).

Visualizando CNNs

Paulo Ricardo Lisboa de Almeida

DSBD

UFPR

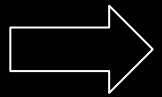
O Impacto das correlações

Considere que remapeamos todos os pixels das imagens usando uma função de remapeamento fixa.

Exemplo:

Em uma imagem em escala de cinza, o pixel na posição A_{00} vai para a posição A_{13}, o pixel na posição A_{12} vai para A_{32}, ...

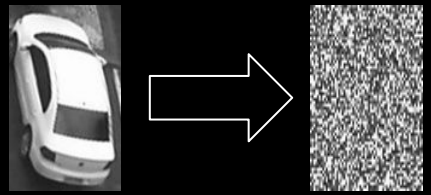
250	248	250	200
189	265	249	221
217	221	199	200
217	221	250	221
217	265	265	221



189	221	221	265
217	250	250	250
248	265	200	221
265	250	249	221
199	217	221	217

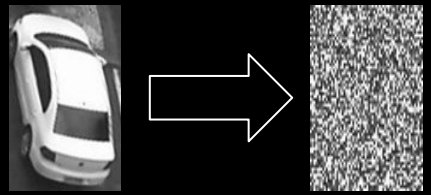
O Impacto das correlações

Aplicando a ideia nas imagens da PKLot.



O Impacto das correlações

Aplicando a ideia nas imagens da PKLot.



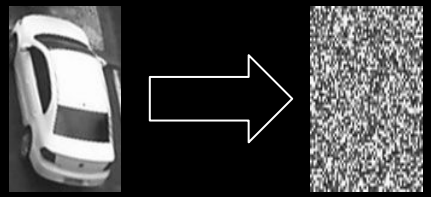
Pergunta:

Ao aplicar o remapeamento em todas as imagens da PKLot, e usando-as para treinar/testar uma CNN e um MLP convencional, como você espera que seja o desempenho dessas redes? Por quê?

Faça você mesmo

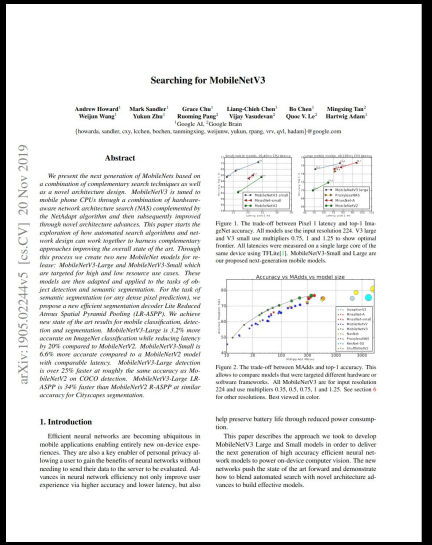
Execute o experimento disponibilizado no Google Colab.

Executa várias vezes, e tire a média dos resultados.



A CNN usada é uma MobileNetV3.

Howard, Andrew, et al. "Searching for mobilenetv3." IEEE/CVF international conference on computer vision. 2019.



Searching for MobileNetV3

Andrew Howard[1] Mark Sandler[1] Grace Chu[1] Liang-Chieh Chen[1] Bo Chen[1] Mingxing Tan[2]
Weijun Wang[1] Yukun Zhu[1] Ruoming Pang[2] Vijay Vasudevan[2] Quoc V. Le[2] Hartwig Adam[1]
[1]Google AI, [2]Google Brain
{howards, sandler, cxy, lcchen, bochen, tanmingxing, weijunw, yukun, rpang, vrv, qvl, hadam}@google.com

arXiv:1905.02244v5 [cs.CV] 20 Nov 2019

Abstract

We present the next generation of MobileNets based on a combination of complementary search techniques as well as a novel architecture design. MobileNetV3 is tuned to mobile phone CPUs through a combination of hardware-aware network architecture search (NAS) complemented by the NetAdapt algorithm and then subsequently improved through novel architecture advances. This paper starts the exploration of how automated search algorithms and network design can work together to harness complementary approaches improving the overall state of the art. Through this process we create two new MobileNet models for release: MobileNetV3-Large and MobileNetV3-Small which are targeted for high and low resource use cases. These models are then adapted and applied to the tasks of object detection and semantic segmentation. For the task of semantic segmentation (or any dense pixel prediction), we propose a new efficient segmentation decoder Lite Reduced Atrous Spatial Pyramid Pooling (LR-ASPP). We achieve new state of the art results for mobile classification, detection and segmentation. MobileNetV3-Large is 3.2% more accurate on ImageNet classification while reducing latency by 20% compared to MobileNetV2. MobileNetV3-Small is 6.6% more accurate compared to a MobileNetV2 model with comparable latency. MobileNetV3-Large detection is over 25% faster at roughly the same accuracy as MobileNetV2 on COCO detection. MobileNetV3-Large LR-ASPP is 34% faster than MobileNetV2 R-ASPP at similar accuracy for Cityscapes segmentation.

Figure 1. The trade-off between Pixel 1 latency and top-1 ImageNet accuracy. All models use the input resolution 224. V3 large and V3 small use multipliers 0.75, 1 and 1.25 to show optimal frontier. All latencies were measured on a single large core of the same device using TFLite[1]. MobileNetV3-Small and Large are our proposed next-generation mobile models.

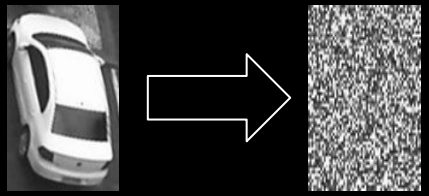
Figure 2. The trade-off between MAdds and top-1 accuracy. This allows to compare models that were targeted different hardware or software frameworks. All MobileNetV3 are for input resolution 224 and use multipliers 0.35, 0.5, 0.75, 1 and 1.25. See section 6 for other resolutions. Best viewed in color.

1. Introduction

Efficient neural networks are becoming ubiquitous in mobile applications enabling entirely new on-device experiences. They are also a key enabler of personal privacy allowing a user to gain the benefits of neural networks without needing to send their data to the server to be evaluated. Advances in neural network efficiency not only improve user experience via higher accuracy and lower latency, but also help preserve battery life through reduced power consumption.

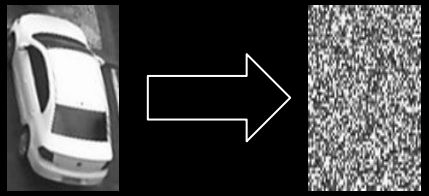
This paper describes the approach we took to develop MobileNetV3 Large and Small models in order to deliver the next generation of high accuracy efficient neural network models to power on-device computer vision. The new networks push the state of the art forward and demonstrate how to blend automated search with novel architecture advances to build effective models.

O Impacto das correlações



	Acurácia no Teste	
	Imagens Originais	Imagens Remapeadas
CNN	94%	82%
MLP	82%	79%

O Impacto das correlações

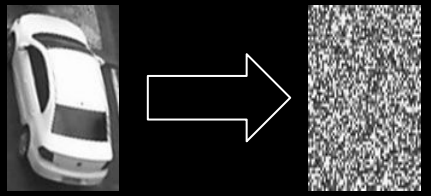


A informação ainda está na imagem, só mudou de lugar!

No entanto, uma CNN espera que haja uma correlação entre pixels vizinhos.

- Essa correlação foi quebrada quando os pixels migraram de lugar.
- É esperada uma queda na acurácia da CNN.

O Impacto das correlações



A informação ainda está na imagem, só mudou de lugar!

No entanto, uma CNN espera que haja uma correlação entre pixels vizinhos.

- Essa correlação foi quebrada quando os pixels migraram de lugar.
- É esperada uma queda na acurácia da CNN.

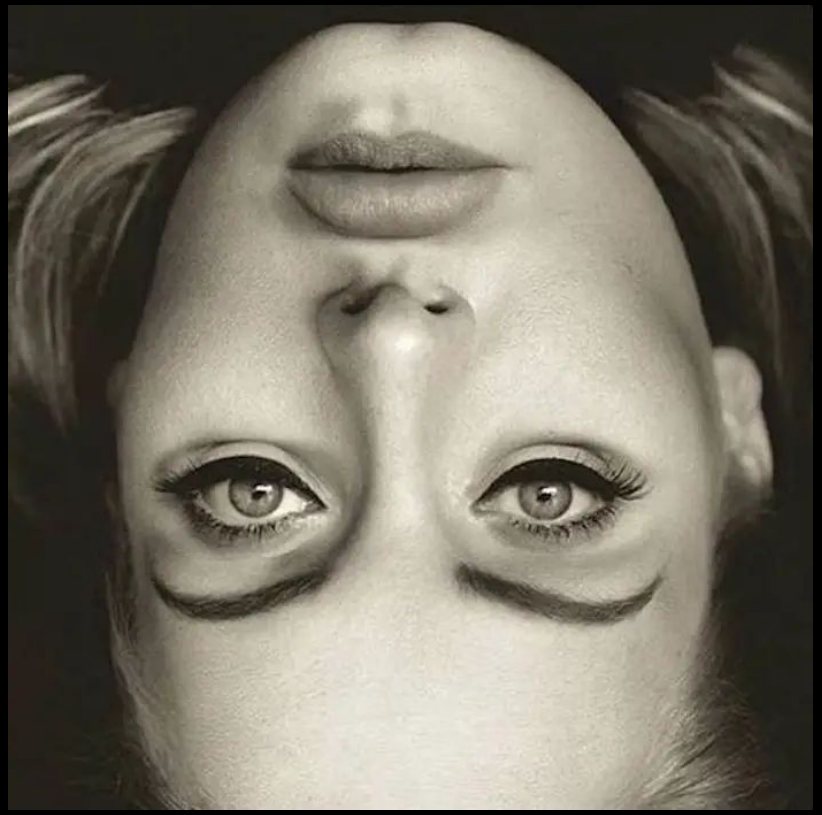
Um MLP *fully connected* não depende da correlação entre pixels vizinhos.

- Para o MLP, é irrelevante a posição dos pixels, desde que o mapeamento seja o mesmo para todas as imagens, durante o treinamento e teste.

Filtros de Convolução

Os filtros de convolução são inspirados no córtex visual humano.

Detecção de orientações em particular e localização de componentes no campo de visão.

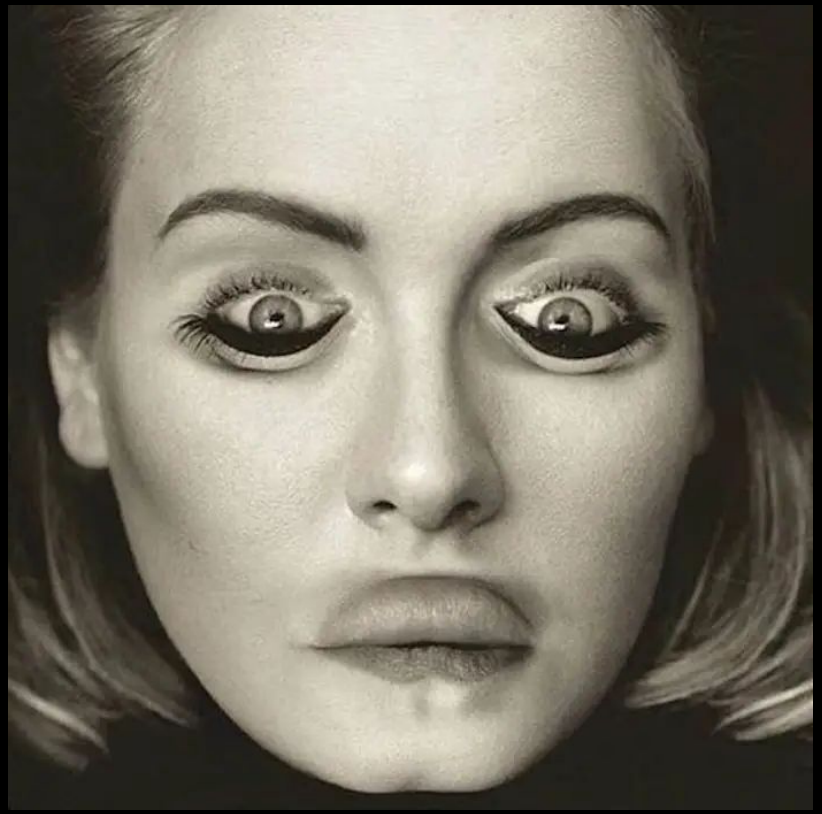


Adele.

Filtros de Convolução

Os filtros de convolução são inspirados no córtex visual humano.

Detecção de orientações em particular e localização de componentes no campo de visão.

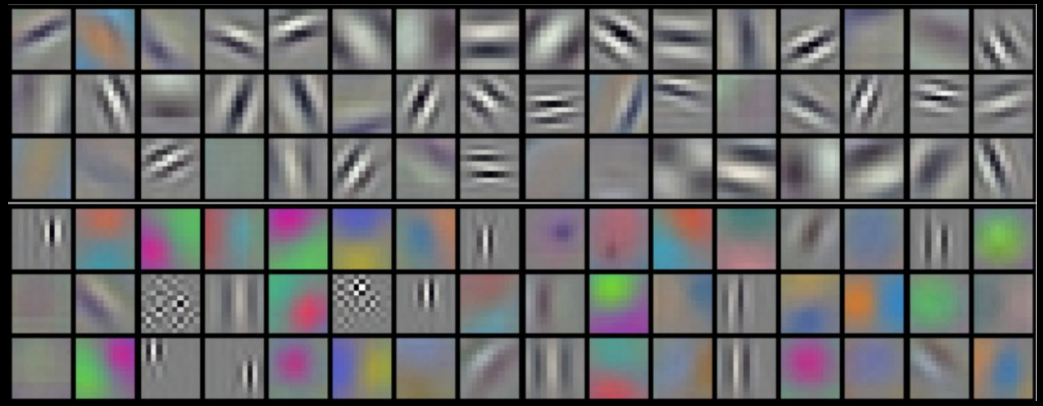


Adele.

Filtros de Convolução

Podemos visualizar os filtros de convolução que rede aprendeu.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*. 2012.

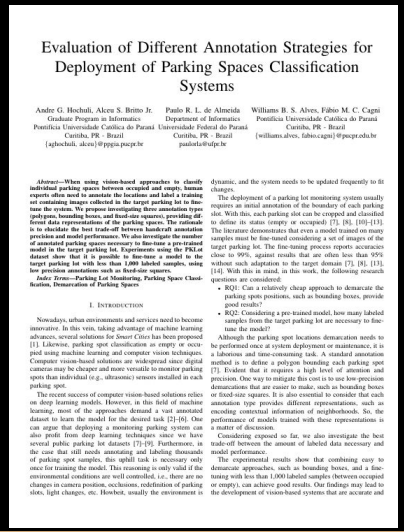


Filtros da primeira camada da AlexNet treinada usando a ImageNet.

Faça você mesmo

Verifique os filtros de convolução que a rede aprendeu no exemplo do Google Colab.

Hochuli, Britto, Almeida, Alves, Cagni.
Evaluation of different annotation strategies for deployment of parking spaces classification systems. IJCNN. IEEE. 2022.



Evaluation of Different Annotation Strategies for Deployment of Parking Spaces Classification Systems

Andre G. Hochuli, Alceu S. Britto Jr.
Graduate Program in Informatics
Pontifícia Universidade Católica do Paraná
Curitiba, PR - Brazil
{aghochuli, alceu}@ppgia.pucpr.br

Paulo R. L. de Almeida
Department of Informatics
Universidade Federal do Paraná
Curitiba, PR - Brazil
paulorl@ufpr.br

Williams B. S. Alves, Fábio M. C. Cagni
Pontifícia Universidade Católica do Paraná
Curitiba, PR - Brazil
{williams.alves, fabio.cagni}@pucpr.edu.br

***Abstract*—When using vision-based approaches to classify individual parking spaces between occupied and empty, human experts often need to annotate the locations and label a training set containing images collected in the target parking lot to fine-tune the system. We propose investigating three annotation types (polygons, bounding boxes, and fixed-size squares), providing different data representations of the parking spaces. The rationale is to elucidate the best trade-off between handcraft annotation precision and model performance. We also investigate the number of annotated parking spaces necessary to fine-tune a pre-trained model in the target parking lot. Experiments using the PKLot dataset show that it is possible to fine-tune a model to the target parking lot with less than 1,000 labeled samples, using low precision annotations such as fixed-size squares.**

***Index Terms*—Parking Lot Monitoring, Parking Space Classification, Demarcation of Parking Spaces**

I. Introduction

Nowadays, urban environments and services need to become innovative. In this vein, taking advantage of machine learning advances, several solutions for *Smart Cities* has been proposed [1]. Likewise, parking spot classification as empty or occupied using machine learning and computer vision techniques. Computer vision-based solutions are widespread since digital cameras may be cheaper and more versatile to monitor parking spots than individual (e.g., ultrasonic) sensors installed in each parking spot.

The recent success of computer vision-based solutions relies on deep learning models. However, in this field of machine learning, most of the approaches demand a vast annotated dataset to learn the model for the desired task [2]–[6]. One can argue that deploying a monitoring parking system can also profit from deep learning techniques since we have several public parking lot datasets [7]–[9]. Furthermore, in the case that still needs annotating and labeling thousands of parking spot samples, this uphill task is necessary only once for training the model. This reasoning is only valid if the environmental conditions are well controlled, i.e., there are no changes in camera position, occlusions, redefinition of parking slots, light changes, etc. Howbeit, usually the environment is dynamic, and the system needs to be updated frequently to fit changes.

The deployment of a parking lot monitoring system usually requires an initial annotation of the boundary of each parking slot. With this, each parking slot can be cropped and classified to define its status (empty or occupied) [7], [8], [10]–[13]. The literature demonstrates that even a model trained on many samples must be fine-tuned considering a set of images of the target parking lot. The fine-tuning process reports accuracies close to 99%, against results that are often less than 95% without such adaptation to the target domain [7], [8], [13], [14]. With this in mind, in this work, the following research questions are considered:

- RQ1: Can a relatively cheap approach to demarcate the parking spots positions, such as bounding boxes, provide good results?
- RQ2: Considering a pre-trained model, how many labeled samples from the target parking lot are necessary to fine-tune the model?

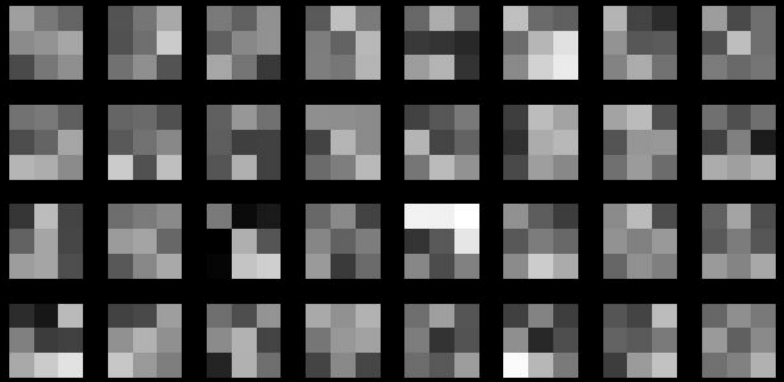
Although the parking spot locations demarcation needs to be performed once at system deployment or maintenance, it is a laborious and time-consuming task. A standard annotation method is to define a polygon bounding each parking spot [7]. Evident that it requires a high level of attention and precision. One way to mitigate this cost is to use low-precision demarcations that are easier to make, such as bounding boxes or fixed-size squares. It is also essential to consider that each annotation type provides different representations, such as encoding contextual information of neighborhoods. So, the performance of models trained with these representations is a matter of discussion.

Considering exposed so far, we also investigate the best trade-off between the amount of labeled data necessary and model performance.

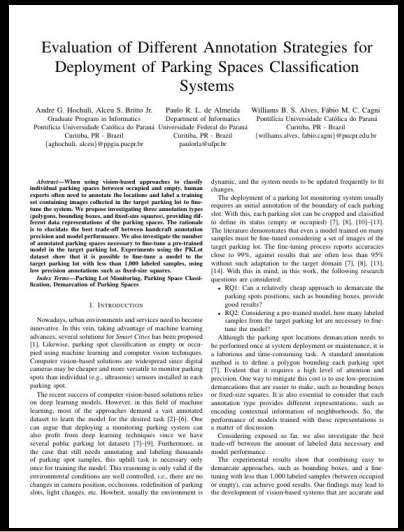
The experimental results show that combining easy to demarcate approaches, such as bounding boxes, and a fine-tuning with less than 1,000 labeled samples (between occupied or empty), can achieve good results. Our findings may lead to the development of vision-based systems that are accurate and

Faça você mesmo

Verifique os filtros de convolução que a rede aprendeu no exemplo do Google Colab.



Hochuli, Britto, Almeida, Alves, Cagni. Evaluation of different annotation strategies for deployment of parking spaces classification systems. IJCNN. IEEE. 2022.



Evaluation of Different Annotation Strategies for Deployment of Parking Spaces Classification Systems

Andre G. Hochuli, Alceu S. Britto Jr.
Graduate Program in Informatics
Pontifícia Universidade Católica do Paraná
Curitiba, PR - Brazil
{aghochuli, alceu}@ppgia.pucpr.br

Paulo R. L. de Almeida
Department of Informatics
Universidade Federal do Paraná
Curitiba, PR - Brazil
paulorl@ufpr.br

Williams B. S. Alves, Fábio M. C. Cagni
Pontifícia Universidade Católica do Paraná
Curitiba, PR - Brazil
{williams.alves, fabio.cagni}@pucpr.edu.br

***Abstract*—When using vision-based approaches to classify individual parking spaces between occupied and empty, human experts often need to annotate the locations and label a training set containing images collected in the target parking lot to fine-tune the system. We propose investigating three annotation types (polygons, bounding boxes, and fixed-size squares), providing different data representations of the parking spaces. The rationale is to elucidate the best trade-off between handcraft annotation precision and model performance. We also investigate the number of annotated parking spaces necessary to fine-tune a pre-trained model in the target parking lot. Experiments using the PKLot dataset show that it is possible to fine-tune a model to the target parking lot with less than 1,000 labeled samples, using low precision annotations such as fixed-size squares.**

***Index Terms*—Parking Lot Monitoring, Parking Space Classification, Demarcation of Parking Spaces**

I. Introduction

Nowadays, urban environments and services need to become innovative. In this vein, taking advantage of machine learning advances, several solutions for *Smart Cities* has been proposed [1]. Likewise, parking spot classification as empty or occupied using machine learning and computer vision techniques. Computer vision-based solutions are widespread since digital cameras may be cheaper and more versatile to monitor parking spots than individual (e.g., ultrasonic) sensors installed in each parking spot.

The recent success of computer vision-based solutions relies on deep learning models. However, in this field of machine learning, most of the approaches demand a vast annotated dataset to learn the model for the desired task [2]–[6]. One can argue that deploying a monitoring parking system can also profit from deep learning techniques since we have several public parking lot datasets [7]–[9]. Furthermore, in the case that still needs annotating and labeling thousands of parking spot samples, this uphill task is necessary only once for training the model. This reasoning is only valid if the environmental conditions are well controlled, i.e., there are no changes in camera position, occlusions, redefinition of parking slots, light changes, etc. Howbeit, usually the environment is dynamic, and the system needs to be updated frequently to fit changes.

The deployment of a parking lot monitoring system usually requires an initial annotation of the boundary of each parking slot. With this, each parking slot can be cropped and classified to define its status (empty or occupied) [7], [8], [10]–[13]. The literature demonstrates that even a model trained on many samples must be fine-tuned considering a set of images of the target parking lot. The fine-tuning process reports accuracies close to 99%, against results that are often less than 95% without such adaptation to the target domain [7], [8], [13], [14]. With this in mind, in this work, the following research questions are considered:

- RQ1: Can a relatively cheap approach to demarcate the parking spots positions, such as bounding boxes, provide good results?
- RQ2: Considering a pre-trained model, how many labeled samples from the target parking lot are necessary to fine-tune the model?

Although the parking spot locations demarcation needs to be performed once at system deployment or maintenance, it is a laborious and time-consuming task. A standard annotation method is to define a polygon bounding each parking spot [7]. Evident that it requires a high level of attention and precision. One way to mitigate this cost is to use low-precision demarcations that are easier to make, such as bounding boxes or fixed-size squares. It is also essential to consider that each annotation type provides different representations, such as encoding contextual information of neighborhoods. So, the performance of models trained with these representations is a matter of discussion.

Considering exposed so far, we also investigate the best trade-off between the amount of labeled data necessary and model performance.

The experimental results show that combining easy to demarcate approaches, such as bounding boxes, and a fine-tuning with less than 1,000 labeled samples (between occupied or empty), can achieve good results. Our findings may lead to the development of vision-based systems that are accurate and

Faça você mesmo

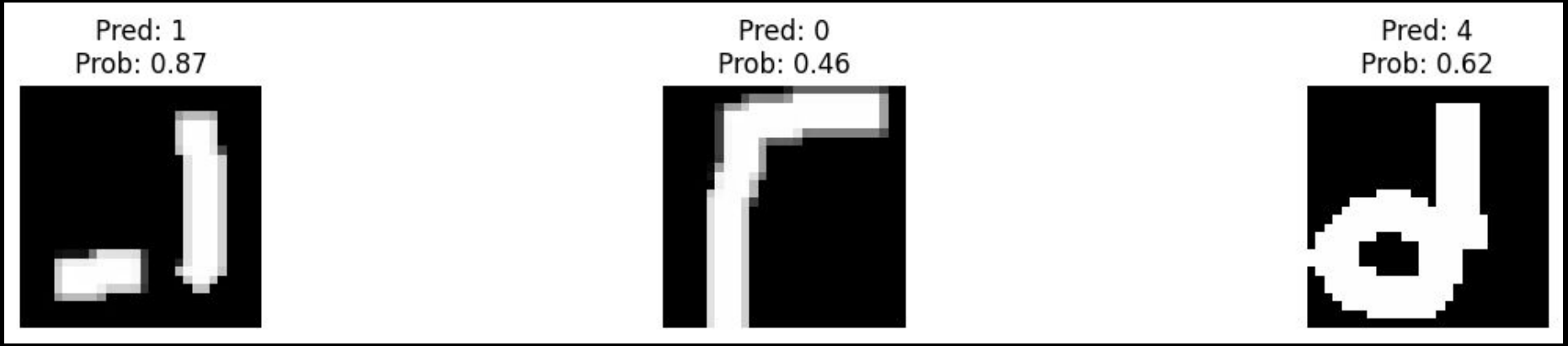
Verifique o exemplo da rede com treinamento na MNist.

Reconhecer dígitos de 0 a 9.

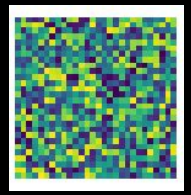
Conclusões

A rede aprende a reconhecer padrões simples com seus filtros.

Exemplo: reconhecer linhas retas como componentes de um número 1.

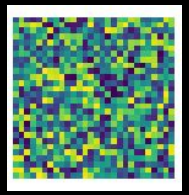


E se mandarmos ruído para a rede?



E se mandarmos ruído para a rede?

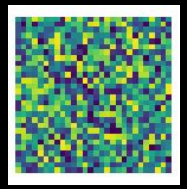
Resultado para a CNN:



Predição: 8 Probabilidade: 0.23.

E se mandarmos ruído para a rede?

Resultado para um MLP:



Predição: 0 Probabilidade: 0.43.

DeepDream

Podemos aplicar uma imagem a uma rede treinada, e calcular o gradiente da imagem em determinada camada da rede.

Modificar os pixels da imagem original de acordo com esse gradiente.

Verificar que tipos de padrões ativam os filtros da rede.

https://en.wikipedia.org/wiki/File:%22Mona_Lisa%22_with_DeepDream_effect_using_VGG16_network_trained_on_ImageNet.jpg

Exercícios

1. Assista a esse vídeo:
   https://www.youtube.com/watch?v=rA5qnZUXcqo&t=1195s

2. Execute esse tutorial do DeepDream:
   https://www.tensorflow.org/tutorials/generative/deepdream?hl=pt-br

Referências

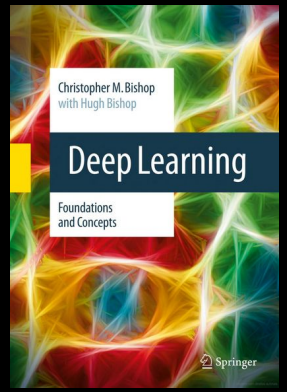
Bishop, C. M., Bishop, H. Deep Learning: Foundations and Concepts. 2023.

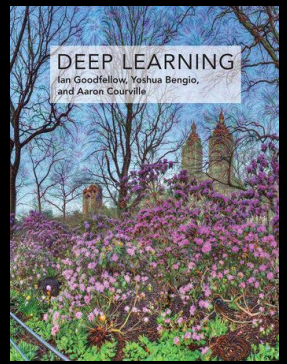
Goodfellow, I., Bengio, Y., Courville, A. Deep Learning. 2016.

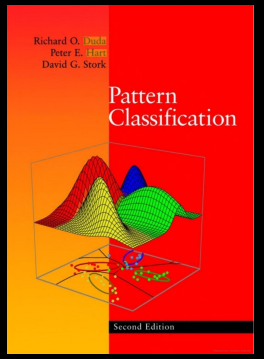
Duda, R. O., Hart, P. E., Stork, D. G. Pattern Classification. 2012.

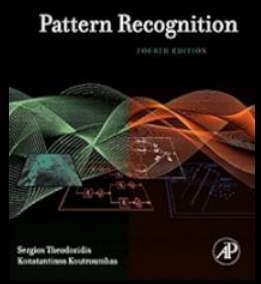
Theodoridis, S., Koutroumbas, K. Pattern Recognition & Matlab Intro. 2010.

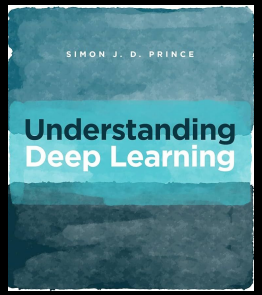
Prince, S. J. Understanding Deep Learning. 2023.











Licença