09

Para saber mais

Datasets no PyTorch

No script `Datasets.ipynb` , conhecemos as aplicações mais comuns para CNNs através dos datasets fornecidos no pacote Torchvision (https://pytorch.org/docs/stable/torchvision/datasets.html) do PyTorch. Vale a pena explorarmos um pouco melhor essa biblioteca de datasets, e a importância desses repositórios de dados.

A primeira dúvida que pode pairar na sua cabeça é:

> *Eu irei trabalhar com meus próprios dados, por que precisaria dos dados do Torchvision?*

Pois saiba que esses datasets são especialmente úteis para auxiliar na solução de outros problemas. As 3 principais utilidades para os datasets clássicos da literatura são:

- **Conhecendo os problemas clássicos**. Como fizemos na nossa aula, os datasets do Torchvision nos ajudam a conhecer os principais problemas de **visão computacional**, a área relacionada a sistemas que tentam compreender informação visual (imagens, vídeos, etc.). Pra quem tá começando agora, carregar e conhecer os datasets disponíveis podem abrir sua cabeça para os diferentes usos de uma CNN.
- **Testando a qualidade de modelos**. Ainda que o seu objetivo seja trabalhar com um dataset muito específico, talvez até privado, é interessante avaliar a sua abordagem de solução em um dataset já consolidado na literatura. Por serem datasets amplamente utilizados, existe uma referência de bom desempenho para esses dados. Caso o problema que deseja resolver se aproxime de alguma das aplicações contidas no Torchvision, vale a pena o test-drive do seu modelo nesses dados.
- **Pré-treinando modelos**. Dentre os datasets do Torchvision, alguns são o que chamamos de *large scale* (grande escala), ou seja, conjuntos com uma quantidade significativa de dados e portanto capazes de treinar modelos mais robustos. Esses dados são ideais para o pré-treinamento de modelos, que se tornam capazes de generalizar para muitos outros problemas através de simples ajustes. Um exemplo muito comum são os modelos treinados no ImageNet que veremos mais à frente nesse post. O Torchvision possui uma biblioteca (https://pytorch.org/docs/stable/torchvision/models.html) com os principais modelos pré-treinados no ImageNet para classificação de imagens.

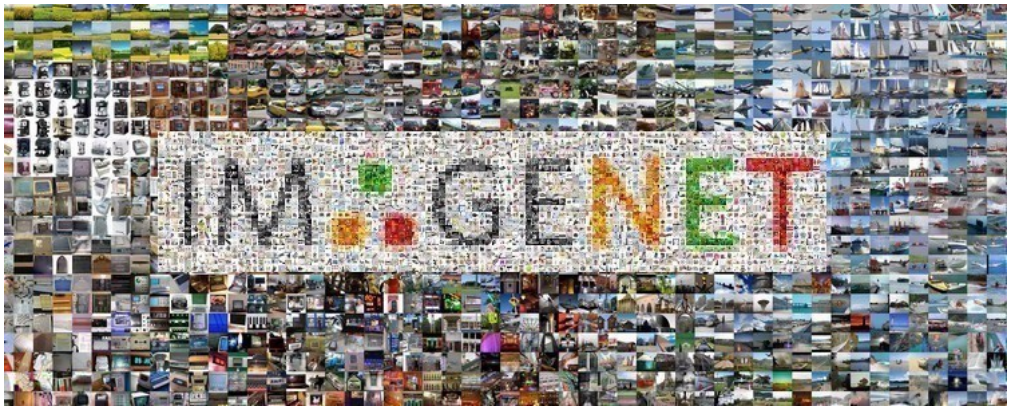
Já conhecemos a ponta do iceberg que representa os problemas com imagens: classificação, detecção e segmentação. O PyTorch traz inúmeros datasets relacionados a esses problemas, vamos conhecer alguns deles a seguir. Além das aplicações clássicas, há uma infinidade de outras informações que podem ser extraídas a partir de imagens. Também comentaremos algumas dessas ao final.

Classificação

- MNIST (https://pytorch.org/docs/stable/torchvision/datasets.html#mnist) e FashionMNIST (https://pytorch.org/docs/stable/torchvision/datasets.html#mnist): Imagens em escala de cinza representando dígitos escritos à mão (MNIST), e peças de roupas (FashionMNIST). Trata-se de um dataset mais simples, pois as informações são apresentadas de forma controlada (centralizada e frontal), além de ter o fundo removido, destacando somente a imagem relevante. Há também variações contendo caracteres japoneses (KMNIST (https://pytorch.org/docs/stable/torchvision/datasets.html#kmnist)) e também letras (EMNIST (https://pytorch.org/docs/stable/torchvision/datasets.html#emnist))

- CIFAR (https://pytorch.org/docs/stable/torchvision/datasets.html#cifar): Um dataset de imagens minúsculas (tiny images) como referido pelos próprios autores. São imagens naturais, ou seja, fotografias do mundo real (com ruído, fundo, variação de luz, etc.), sendo portanto muito mais desafiador que o MNIST. Possui duas variações: CIFAR10, com 10 categorias de objetos e CIFAR100, com 100 categorias.

- ImageNet (https://pytorch.org/docs/stable/torchvision/datasets.html#emnist): Um dos datasets mais importantes da história da visão computacional, traz mais de um milhão de imagens anotadas que variam entre 1000 categorias de objetos. Modelos treinados nesse dataset são capazes de generalizar para outros problemas, como já extensivamente demonstrado por pesquisadores. O PyTorch inclusive traz uma biblioteca (https://pytorch.org/docs/stable/torchvision/models.html) com os principais modelos de classificação de imagens pré-treinados no ImageNet.



Além de objetos, o Torchvision tem também datasets para classificação de cenas (LSUN (https://pytorch.org/docs/stable/torchvision/datasets.html#lsun)) e imagens de celebridades (CelebA (https://pytorch.org/docs/stable/torchvision/datasets.html#celeba)) com múltiplos rótulos de classificação, como a identidade dos indivíduos ou atributos do rosto (franja, óculos, etc.).

Detecção

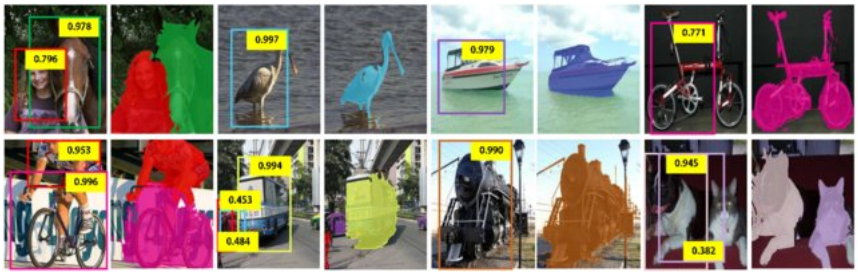
Em geral, datasets de detecção e segmentação oferecem também o rótulo com a categoria daquele elemento, por ser uma anotação mais simples de ser adquirida (apenas um identificador, um número, que representa a imagem inteira). O contrário não é verdade, poucos datasets de classificação oferecem rótulos com maior nível de detalhe como *bounding boxes* para detecção ou máscaras para segmentação.

- SVHN (https://pytorch.org/docs/stable/torchvision/datasets.html#svhn): Apesar deste ser um dataset majoritariamente de classificação, decidi trazê-lo para esta seção pois ele também fornece rótulos de detecção. Da mesma forma que o MNIST original, este também é um dataset de classificação de dígitos, com um diferencial: são imagens naturais provenientes do Google *Street View*. A imagem a seguir mostra exemplos das imagens acompanhadas dos rótulos de detecção.

- CocoDetection (https://pytorch.org/docs/stable/torchvision/datasets.html#detection): COCO (*Common objects in context*, objetos comuns em contexto) é uma competição que fornece um dataset de grande escala com múltiplas modalidades de rótulo, dentre eles a detecção. A página do desafio (http://cocodataset.org/#detection-leaderboard) traz inclusive uma tabela de classificação com o desempenho dos melhores participantes, servindo de referencial para novas metodologias.

- PascalVOC - Detection (https://pytorch.org/docs/stable/torchvision/datasets.html#torchvision.datasets.VOCDetection): O Pascal *Visual Object Classes* também foi uma popular competição com múltiplas modalidades que consolidou um dos principais

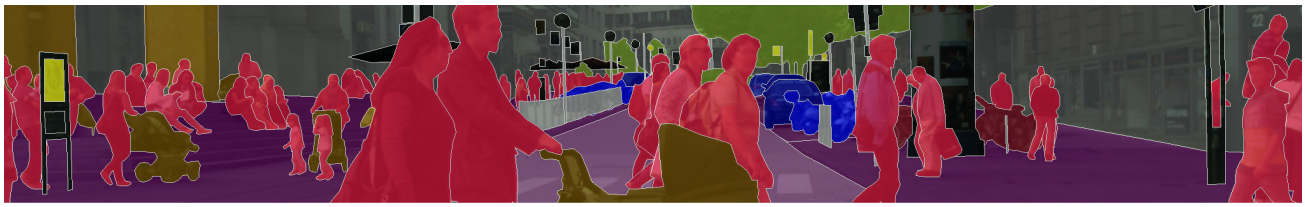
datasets de imagem. A imagem a seguir demonstra resultados de classificação e segmentação de um trabalho acadêmico.



Segmentação

Como vimos na seção de detecção, os datasets COCO e PascalVOC também oferecem rótulos de segmentação, no entanto o Torchvision fornece esse tipo de rótulo somente para o PascalVOC (https://pytorch.org/docs/stable/torchvision/datasets.html#torchvision.datasets.VOCSegmentation). Além dele, há também um dataset muito importante para a área de segmentação:

- Cityscapes (https://pytorch.org/docs/stable/torchvision/datasets.html#cityscapes): contendo imagens de cenas urbanas, este dataset oferece uma variedade robusta de imagens em diferentes cidades europeias, com diferentes condições de clima. Ele oferece rótulos para os objetos mais comuns nesse tipo de cena, como árvores, humanos, veículos, placas de trânsito, entre outros. Visite a página do dataset (https://www.cityscapes-dataset.com/dataset-overview/#class-definitions) para maiores detalhes.



Note que quanto mais denso o rótulo (classe, bounding box, máscara de segmentação), mais escassa é a quantidade de datasets e amostras por dataset, dada a maior dificuldade de produzir tais rótulos.

Outras aplicações

Mas há também problemas como ***Image Captioning*** (legendagem de imagens), que mistura o uso de CNNs com outro tipo de rede, responsável pela geração do texto da legenda. Veja na imagem a seguir um exemplo do dataset CocoCaption (https://pytorch.org/docs/stable/torchvision/datasets.html#captions), também encontrado no Torchvision. Nessa mesma categoria de problemas, o Torchvision traz também o SBU (http://www.cs.virginia.edu/~vicente/sbucaptions/).

The man at bat readies to swing at the pitch while the umpire looks on.

A large bus sitting next to a very tall building.

Uma parte importante (e difícil) da visão computacional é o trabalho com vídeos. Um dos principais desafios nessa área é o **Reconhecimento de ações**. É um problema de classificação para vídeos, onde o rótulo faz referência à principal ação sendo realizada no vídeo. Para essa aplicação, o Torchvision possui três datasets:

- Kinetics-400 (https://pytorch.org/docs/stable/torchvision/datasets.html#kinetics-400)
- HMDB51 (https://pytorch.org/docs/stable/torchvision/datasets.html#hmdb51)
- UCF101 (https://pytorch.org/docs/stable/torchvision/datasets.html#ucf101)

Os números no nome de cada dataset fazem referência à quantidade de categorias de ação anotadas. A imagem a seguir apresenta exemplos retirados dos três datasets.

Faça bom proveito dessa rica biblioteca de dados! Seja para pré-treinar modelos, estudar ou experimentar uma nova abordagem, vale a pena recorrer primeiro para bases de dados já consolidadas.