## Topics

**Find inspiration at Kaggle** [**https://www.kaggle.com/**](https://www.kaggle.com/)

**Semantic Image Segmentation**

For every pixel of an image, predict to which object it belongs to. One image can contain multiple objects. [PASCAL VOC 2009](http://host.robots.ox.ac.uk/pascal/VOC/voc2009/index.html#data)

Blog: [Transfer Learning for Segmentation Using DeepLabv3 in PyTorch](https://towardsdatascience.com/transfer-learning-for-segmentation-using-deeplabv3-in-pytorch-f770863d6a42)

Packages: <https://github.com/qubvel/segmentation_models.pytorch>

**Handwritten Character Recognition**

Recognize 62 different symbols (10 digits, 26 lowercase alphabet characters, 26 uppercase alphabet characters). The dataset contains 814 255 28x28 images. [EMNIST](https://www.nist.gov/itl/products-and-services/emnist-dataset)

**PASCAL VOC classification (imbalanced dataset)**

The task is to classify 20 different objects in an image (224x224). Every image can contain multiple objects (but for every image you can select just one object, so then every image would have just one label). [PASCAL VOC 2009](http://host.robots.ox.ac.uk/pascal/VOC/voc2009/index.html#data)

**Facial Keypoints Detection (medium)**

The objective of this task is to predict 15 keypoint positions on face images (left\_eye\_center, right\_eye\_center, mouth\_center\_top\_lip, …). Facial keypoint localization is a key step in several applications, such as analysing facial expressions and face recognition.

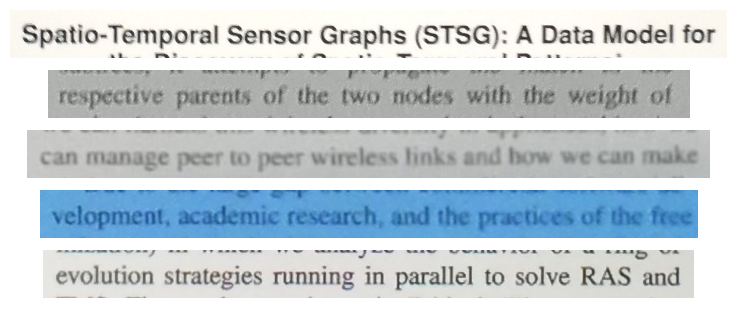
Resources: [Kaggle competition](https://www.kaggle.com/c/facial-keypoints-detection/overview)

**Text Recognition (OCR) (bit advanced)**

Given a cropped image of a text line, transcribe its text. You can use a convolutional+recurrent neural network with a  [CTC loss](https://pytorch.org/docs/stable/generated/torch.nn.CTCLoss.html).

Info: [Multi-Digit Sequence Recognition With CRNN and CTC Loss Using PyTorch Framework](https://medium.com/swlh/multi-digit-sequence-recognition-with-crnn-and-ctc-loss-using-pytorch-framework-269a7aca2a6)

Dataset: [**Brno Mobile OCR Dataset**](https://pero.fit.vutbr.cz/brno_mobile_ocr_dataset) - [Cropped text lines with transcriptions](https://www.fit.vutbr.cz/~ikiss/b-mod/b-mod_lines.zip) (5.29 GB)



### NLP topics:

**Unsupervised Topic Modeling**

Tools for unsupervised topic discovery are commonly used by NLP practitioners to understand what the corpus data are about.

* Discover what are the (e.g. 10) dominant topics inside the data corpus.
* Find what is the most dominant topic.
* What are the words most commonly used within each topic?
* Plot word-cloud to illustrate contents of each topic.
* What are the most representative documents for each topic?
* Plot 2D T-SNE representations of all documents according to their topic affiliation.

Dataset: [Newsgroups](https://raw.githubusercontent.com/selva86/datasets/master/newsgroups.json)

Tutorial source: [Blogpost](https://www.machinelearningplus.com/nlp/topic-modeling-visualization-how-to-present-results-lda-models/)

**Sentiment Classification**

Classify language’s sentiment.

Dataset: [SST](https://nlp.stanford.edu/sentiment/)

Tutorial source: [Blogpost](https://towardsdatascience.com/fine-grained-sentiment-analysis-in-python-part-1-2697bb111ed4)

Sentiment classes in SST-5:

very negative, negative, neutral, positive, very positive (see README.md from dataset to understand how to split dataset into 5 categories)

**Question Paraphrase classification**

Binary classification of question pair duplicity. Given two questions, decide whether the questions are paraphrases of each other.   
1 if *question\_1* and *question\_2* have essentially the same meaning, and 0 otherwise

Dataset: [QQP](https://www.kaggle.com/c/quora-question-pairs)

Tutorial source: Kaggle (see dataset)

**Natural Language Inference**

Given premise and hypothesis, judge whether hypothesis:

* **contradicts** the statement in premise,
* **entails** the statement in premise,
* or is **neutral** towards the statement in premise

Dataset: [SNLI](https://nlp.stanford.edu/projects/snli/)

Tutorial source: This is essentially a task similar to QQP, but the classification is made into 3 classes. Thus if you need help, utilize QQP tutorial sources and change the dataset to SNLI.

Examples:

| **Premise** | **Judgments** | **Hypothesis** |
| --- | --- | --- |
| A man inspects the uniform of a figure in some East Asian country. | contradiction | The man is sleeping |
| An older and younger man smiling. | neutral | Two men are smiling and laughing at the cats playing on the floor. |
| A soccer game with multiple males playing. | entailment | Some men are playing a sport. |

### Speech topics:

**Speech command classification**

Recognize speech commands from audio. Input is ~1 second long audio containing spoken command such as "down", "go", "left", etc. The task is to recognize the spoken command.

Dataset: [Speech commands](https://arxiv.org/abs/1804.03209)

Tutorial: [Tensorflow](https://www.tensorflow.org/tutorials/audio/simple_audio) / [PyTorch](https://pytorch.org/tutorials/intermediate/speech_command_recognition_with_torchaudio.html)

**Music Genres Classification**

Given an audio file, classify it into categories corresponding to style (genre) of the music. Could work on hand-designed features or could be a convolutional and recurrent network classifier.

Dataset: [GITZAN dataset](http://opihi.cs.uvic.ca/sound/genres.tar.gz), [GITZAN on Kagle](https://www.kaggle.com/andradaolteanu/gtzan-dataset-music-genre-classification)

Tutorial: [Music Genres Classification](https://www.analyticsvidhya.com/blog/2021/06/music-genres-classification-using-deep-learning-techniques/)

**Audio datasets:** <https://towardsdatascience.com/a-data-lakes-worth-of-audio-datasets-b45b88cd4ad>

**Vision datasets:** <https://public.roboflow.com/>

## [Urban Sound Classification](https://www.kaggle.com/pavansanagapati/urban-sound-classification)

Mozilla Common Voice

TEDLium

Reinforcement Learning

Genetic programming / Genetic Cartesian Programming