Applied AI

brought you by:
inzva, Bev Foundation

Guides:
Halil Eralp Koçaş
Şafak Bilici & Enes Sadi Uysal
Ahmet Melek
Uğur Ali Kaplan
Duygu Ay

This Week: Computer Vision

- Five Problems:
  - Classification +
  - Localization +
  - Segmentation +
  - Object Detection +
  - Image Generation

- Four model types:
  - Dense (Fully-Connected) Networks +
  - Convolutional Neural Networks +
  - Autoencoders
  - Generative Adversarial Networks

- Four Frameworks
  - Tensorflow 2 +
  - Keras Functional +
  - Keras Sequential +
  - Pytorch & torchvision +

- Six Datasets:
  - MNIST +
  - CIFAR-10 +
  - Kaggle Face Keypoint Detection +
  - Segmentation +
  - COCO for Object Detection +
  - Occlusion Dataset

Today’s Schedule

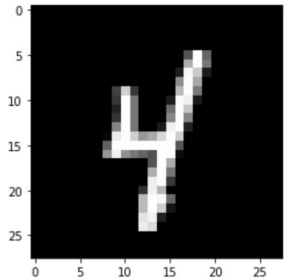
● Introduction	10.00 - 10.15
● MNIST Classification with Dense Nets on Tensorflow 2	10.15 - 10.45
● Similarity Learning on MNIST with Siamese Conv Networks on Keras Functional	10.45 - 11.15
● CIFAR-10 Classification and Transfer Learning with Conv Nets on Keras Sequential	11.15 - 12.00
● **Lunch Break**	**12.00 - 12.45**
● Kaggle Facial Keypoints Detection with Conv Nets on PyTorch	12.45 - 13.25
● Segmentation with U-Net Architecture	13.25 - 13.55
● Segmentation with a Pre-Trained model from Torchvision	13.55 - 14.20
● Object Detection with a Pre-Trained model from Torchvision	13.55 - 14.20
● **Break**	**14.20 - 14.35**
● Homework Description	14.35 - 15.00

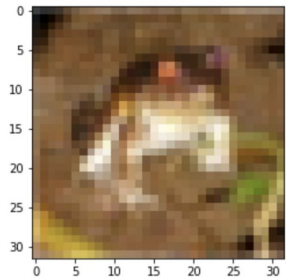
Computer Vision

- Humans can retrieve basic information and many more from looking an image.
  - Four people, many cars, street, etc.
  - People are walking, one of them is barefoot, etc.
  - Their clothes, danger of getting hit by a car, etc.
- Thus, we aim to reach at least this level using Computer Vision algorithms.
- A system must recognize these examples we talked above and more to provide a description as complete as possible of the image.

Classification

- Classification of a given image as belonging to one of a set of predefined categories.





- MNIST Dataset:
  - Handwritten digit classification
  - 60000 training, 10000 test examples
  - Supervised problem
- CIFAR-10 Dataset:
  - Image classification
  - 50000 training, 10000 test examples
  - 32x32x3 images, supervised problem
  - 10 classes, 5000 training, 1000 test each
- CIFAR-100 Dataset:
  - Same settings as CIFAR-10
  - 500 training and 100 test per class
  - 100 classes

Tensorflow 2

- API Cleanup
- Eager execution
  - TF1 requires manually construction of an abstract syntax tree by making API calls.
  - TF2 more Python-like.
- Functions, not sessions
  - TF1 requires of session.run() to execute functions.
  - TF2 more like Python calls -> f(input), etc.
- Keras as a high-level API

Similarity Learning with Siamese Neural Networks

- Goal: understand whether two inputs are same or not.
  - Understanding **similarity**!
- Classification for pre-defined number of classes does not scale.

- Siamese NNs:
  - Why calling Siamese?
    - Same weights are shared for inputs.
  - Process two images in parallel
  - Decide based on a distance function
  - Loss functions:
    - Classification losses work
      - Binary cross-entropy
    - Triplet loss

Keras Functional

- Create more flexible models than Keras Sequential!
  - Functional API can handle models with non-linear topology, shared layers, and multiple inputs & outputs.
- DL models -> Directed Acyclic Graphs (DAG). Functional is a way to build graphs of layers.
- Training, evaluation, inference, and saving models are exactly same for both Functional and Sequential API.
- Can use same layers for multiple models!

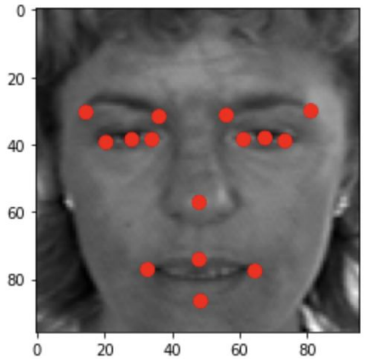
Keras Sequential

- Use when a model is appropriate for **a plain stack of layers** where each layer has **exactly one input tensor and one output tensor**.
- Once a Sequential model is built, it behaves like Functional API model.
  - Each layer has input and output.
  - Can create a new model to observe the output of each layer, etc.

Localization

- Goal is to learn where the objects are in given images.
- Standard way to identify localization in images using bounding boxes to encapsulate localized objects.

- Kaggle Facial Keypoints Dataset
  - Predict keypoint positions on face images.
  - 7049 training, 1783 test images
  - 96x96x1 images
  - Supervised problem
  - 2D 15 features

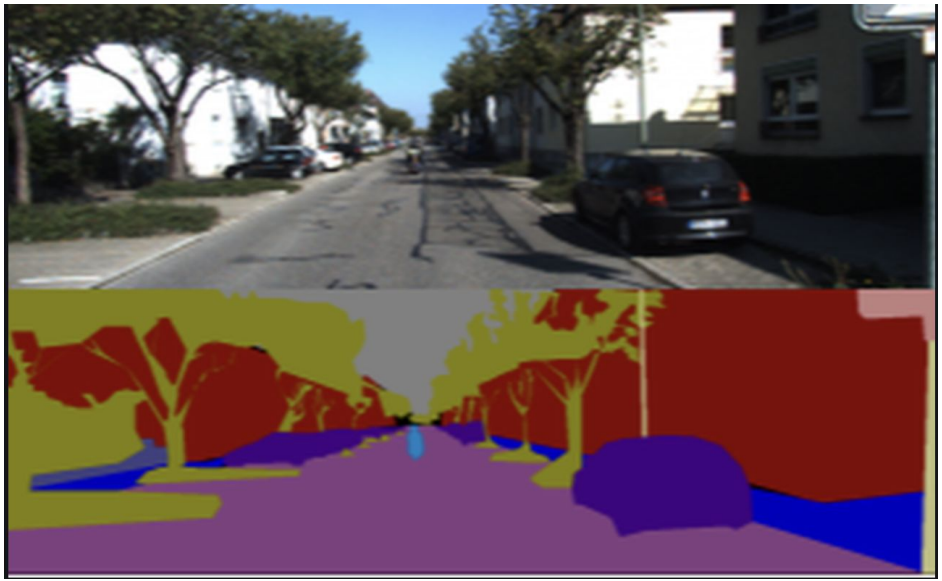


PyTorch

- Optimized tensor library for deep learning using GPUs and CPUs.
- Many libraries are part of PyTorch project for many tasks such as audio, text, vision, etc.
- Ease-of-use and flexible.

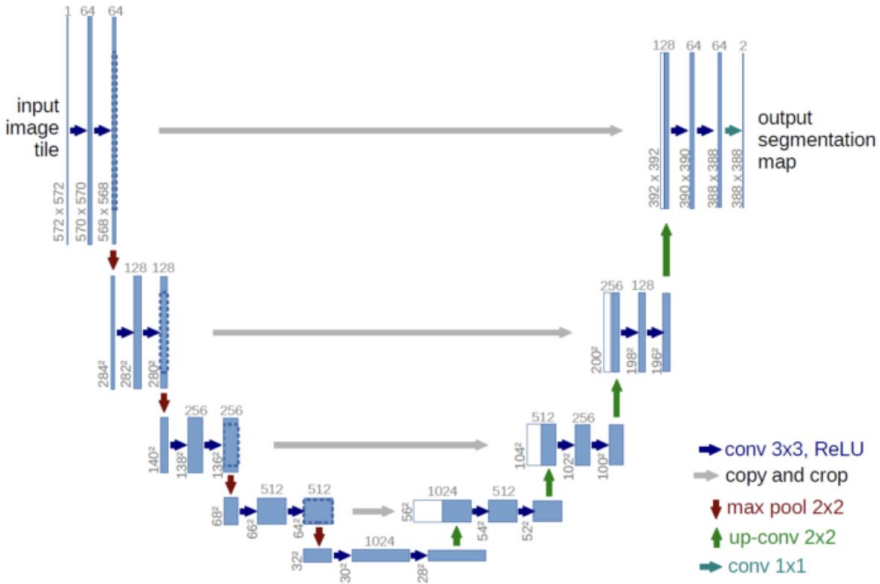
Semantic Segmentation

- Goal: predict class of each pixel in image.
- Output: generally a high resolution output image with same shape of input image with each class is represented with a color.



U-Net Architecture

- Encoder-decoder architecture with skip-connections.
  - Conv layers
  - Max pool layers
  - Up-conv layers
  - Skip-connection layers
- Bottleneck layer for high-level information.
- Skip-connections are for low-level information.
  - Where you utilize depends on use case.



Torchvision

- Part of PyTorch project.
- The torchvision package consists of popular datasets, model architectures, and common image transformations for computer vision.

Object Detection

- Localization and classification of all objects present in a given image.
- Can be multiple classes.
- Bounding boxes to localize and classification to understand what object is in the bounding box.
- Complex Problem
    - Some objects are occluded, partially visible, etc.

- COCO Dataset
    - There are images for all tasks
        - Object detection, segmentation, etc.
    - 330k images (>200k labeled)
    - 80 object, 91 stuff categories