Udacity nanodegree overview

Kurbakov Dmytro

June 9, 2019

Overview

Nanodegree.. what does it mean?

Definition

A Udacity Nanodegree Program is a unique online educational offering designed to bridge the gap between learning and career goals. (Udacity)

Set of the learning courses with a focus on the practical side (Kurbakov D.).

Who is a teacher?

Short answer: Partners with Udacity Full time employees.
Partners;

- Amazon
- AT&T
- Facebook
- Google
- IBM
- Kaggle
- Lyft
- NVidia
- ...

What nanodegreeas are available?

Short answer: a lot.

The most interesting:

- Machine Learning Engineer
- Deep learning
- Deep reinforcement learning
- Self-driving car Engineer
- Artificial Intelligence
- Flying car and autonomous flying engineer
- Robotics software engineer
- Computer vision
- Cloud dev ops engineer
- Cloud developer

How the nanodegree structured?

Core curriculum:

- Each of the curriculum has N parts (topics).
- The part can be optional.
- Non-optional part has the project, that is required to pass, if you want to graduate.
- Non-optional part can have an optional project.
- Only 1 deadline.

Extracurriculum:

- Each of the curriculum has N parts (topics).
- All parts are optional
- No projects
- No deadlines

Community

Once you enroll into the course:

- Access to all courses and projects
- Access to the slack
- Slack channel with everyone who is doing the nenodegree with you
- Assigned to you mentor (once a weekly group call, once a week 1:1 call)
- Access to the career coach
- All project reviwed by human

Once you graduate:

- Access to the alumni portal

Price

Subscription modell
339 Euros per month

Computer vision nanodegree overview

Overview

1. Created in cooperation with NVidian and Affectiva.
2. Scheduled for 3 months with workload 10-15 hours per week.
3. 24 people on the slack group + mentor.

Mentor

Ph.D. in Electrical and Electronical Engineering
Sr. Software Engineer in LG Electronics, located in San Jose, CA, USA.
Working on the simulator for the testing autopilot (tracks).

Course structure: Core curriculum

Sections:

1. Introduction to the computer vision
2. Optional: cloud computing
3. Advanced computer vision and deep learning
4. Object tracking and localization

Introduction to the computer vision

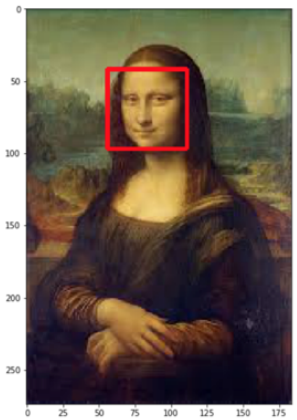
1. RGB, HSV, HLS representation of the image
   - Coding blue/green screen
   - Detecticting day and night on the image
2. Convolutional filters
   - Detecting horisontal/vertical lines
   - Detectig corners on the image
3. ML in computer vision
   - CNN for face feature detection

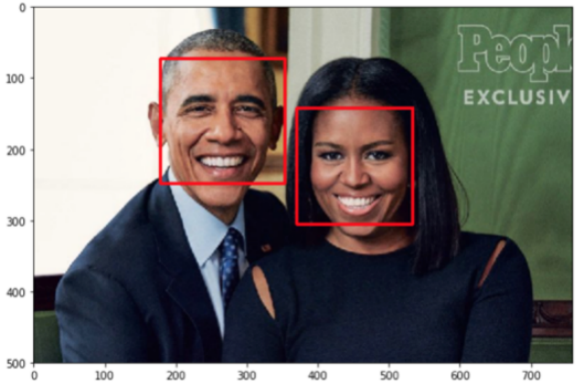
Project: Facial keypoints detection

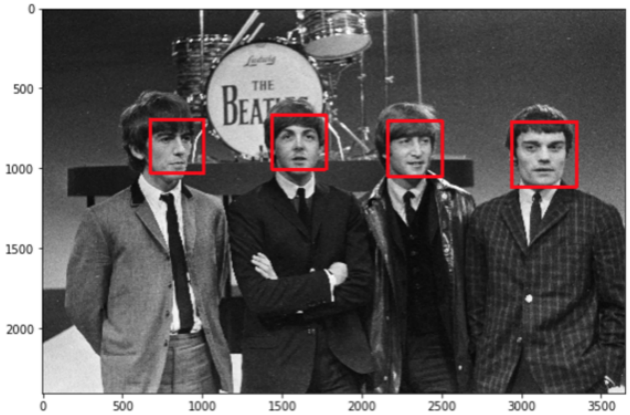
Python + PyTorch + Jupyter Notebook
Input:

Project: Facial keypoints detection

Face detection:

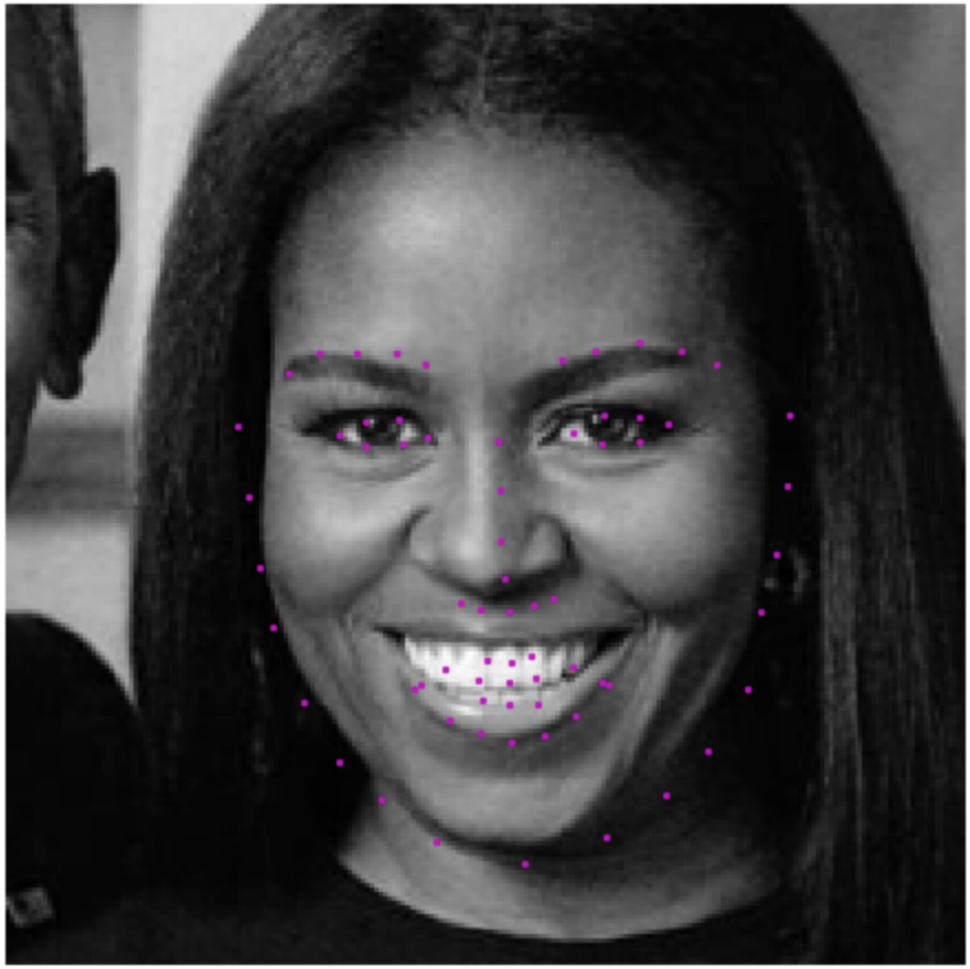






Project: Facial keypoints detection

Output:

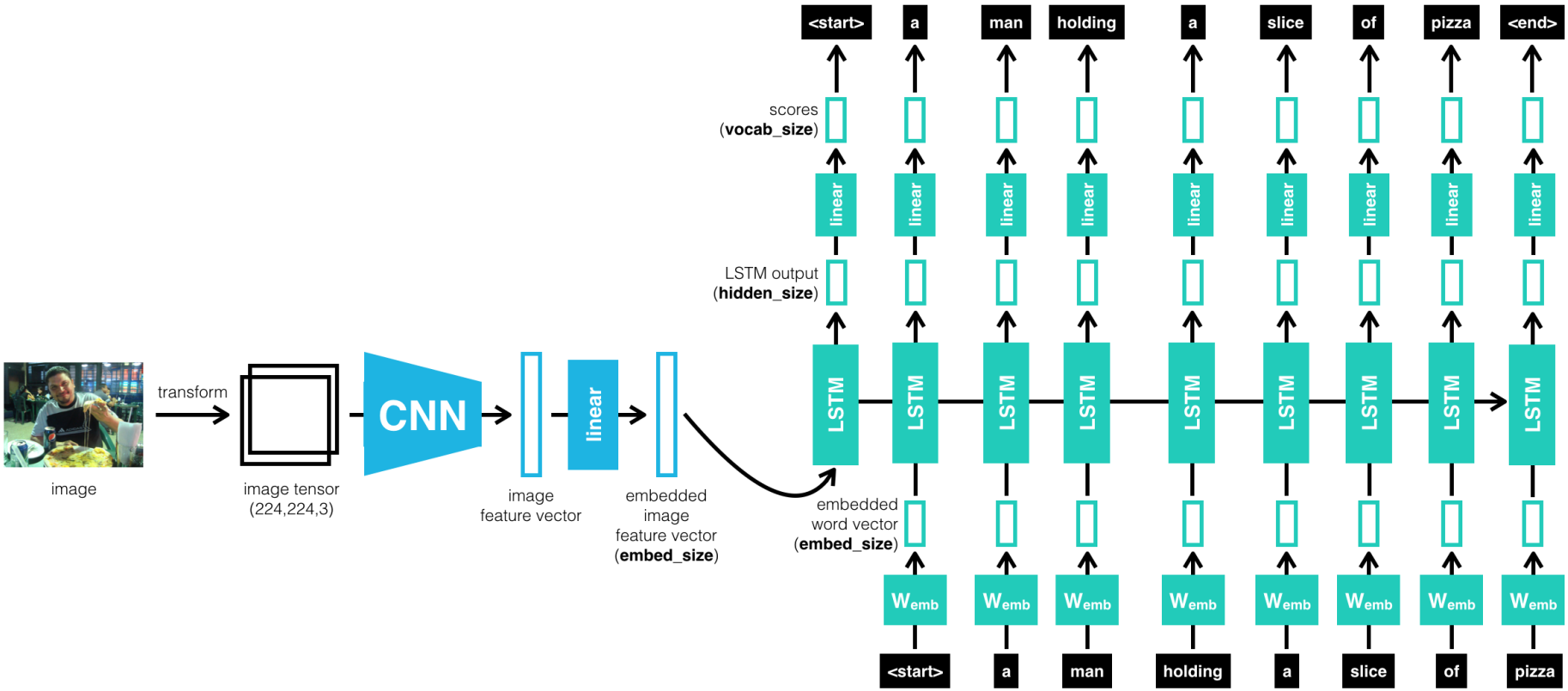


Advanced compuer vision and deep learning

1. YOLO
   You only look once: network for the real-time object detection
   Official webpage, Demo
2. RNN
   Recurrent neural network: understanding the context of the image
3. LSTM
   Long-short-term memory: one of the type of the RNN. This network is used to analize the context

Project: Image captioning

Python + PyTorch + Jupyter Notebook + COCO data set + CUDA



Object tracking and lockalization

- Undestanding of motion
  Methematical representation of the motion, programm 2d map and detect objects with the motion based on the probabilities.
- Kalman filter
  is an algorithm that uses a series of measurements observed over time, containing statistical noise and other inaccuracies, and produces estimates of unknown variables.
  A common application is for guidance, navigation, and control of vehicles.

Project: Landmark detection and tracking (SLAM)

Python + Jupyter notebook

Highly artificail project, that has nothing to do with the CV

The task of the project:

1. Programm the world, where the robot will navigate
2. Simulate each movement of the robot
3. Add some noise
4. Predict the locaion of the robot

Conclusions

Conclusions

What I liked:

1. Was new for me to work with RNN and PyTorch.
2. Better understanding of the image pre-processing.

What I would change?

1. More teoretical stuff
2. More classic CV algorithms
3. Remove SLAM part

Would I recommend the nanodegree? Yes!

Will I do another one? Yes

If you want to do ML, DL, AI or similar, strongly recommend to have a NVidia GPU at home.

Q&A

Questions?