1stDayHack

MENTOR WORKSHOP

Prepared by Keifer Lee

ABOUT ME

My name is Keifer Lee.

Background

| 1stDayHack Organizing Committee Member

| Research Engineer @ Taylor's Unmanned Aerial Vehicle (TUAV) research group

| Current work focuses primarily on Computer Vision for UAV - object tracking

| Machine Learning Research Intern @ 1337 Ventures Malaysia

| AI Engineer @ FourFang

| Degree in Electrical and Electronics Engineering

Area of Interests

Machine Intelligence, Meta-Learning, Computer Vision.

THE OUTLINE

3 MAIN TOPICS

I. Getting started with Python on Kaggle (Session 1 only)

Introduction to Jupyter-Notebooks |

Basics of Python and Key Libs |

Simple Sample Project |

II. Basics of Machine and Deep Learning

Outline and Key Components |

Introduction to PyTorch - DL Library |

Building Simple Classifier on CIFAR-10 |

III. 1stDayHack Mechanics

Purpose and Desired Outcome |

Rules, Mechanic and Themes |

Introduction to 1stDayKit (FDK) |

Recommendations for Mentor |

IV. How to Get Started?

V. Additional Resources

Preface and Setting Up Dependencies

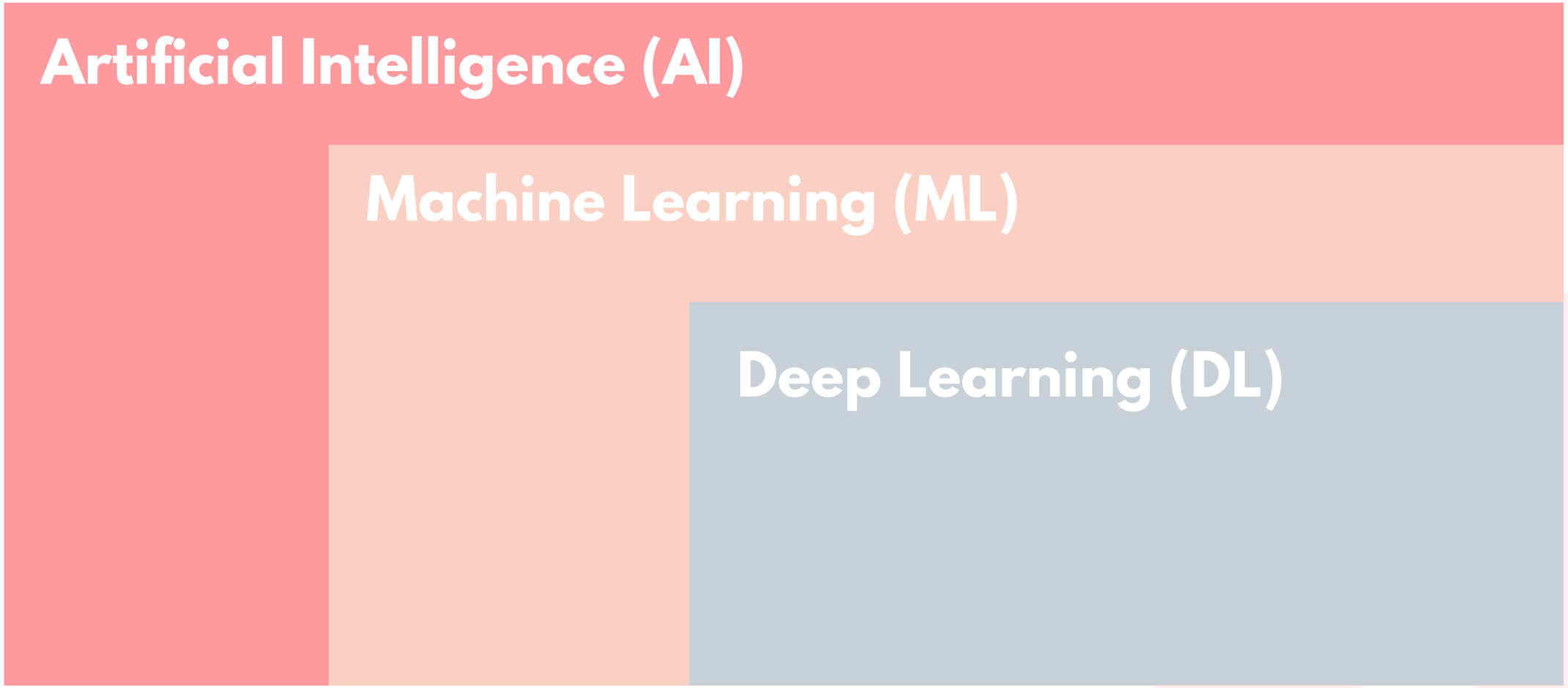
1. Clone the following repo - https://github.com/1stDayHack/lessons
2. Got to Kaggle's website and create an account - www.kaggle.com
3. Go to Kaggle's Kernal page.
4. Create a new Notebook.
5. Import the notebook "part1_python.ipynb" from part1_python folder.
6. Empty your bladders.
7. Sit tight.

That's it!

Part 1: Getting Started with Python on Kaggle

(Move and Continue on Part 1 Notebook)

Part 2: Basics of Machine and Deep Learning


Artificial Intelligence (AI)
Machine Learning (ML)
Deep Learning (DL)

Why Now?

Neural Networks date back decades, so why the resurgence?

1952 Stochastic Gradient Descent

1958 Perceptron
- Learnable Weights

1986 Backpropagation
- Multi-Layer Perceptron

1995 Deep Convolutional NN
- Digit Recognition

1. Big Data

- Larger Datasets
- Easier Collection & Storage

IMAGENET

WIKIPEDIA
The Free Encyclopedia

2. Hardware

- Graphics Processing Units (GPUs)
- Massively Parallelizable

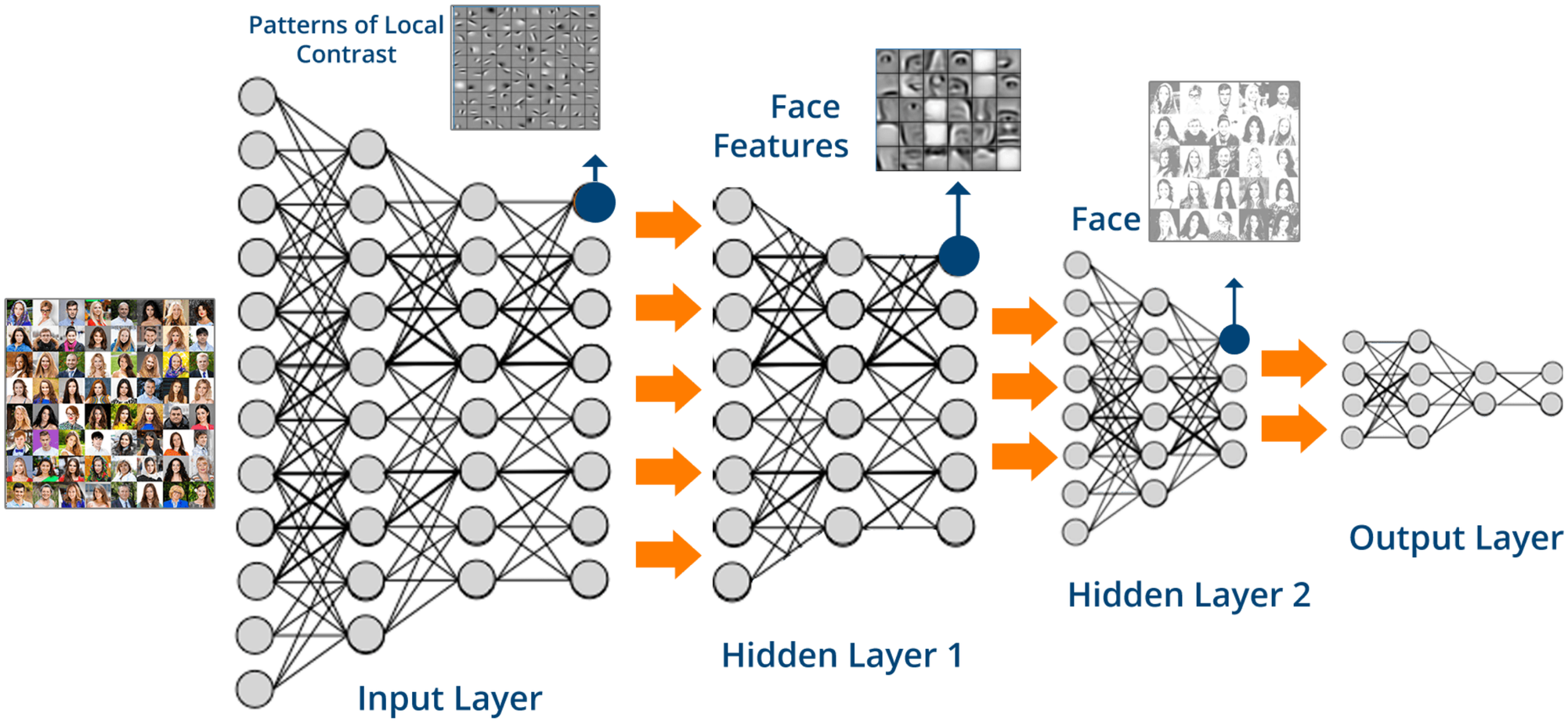
3. Software

- Improved Techniques
- New Models
- Toolboxes

TensorFlow

Bread & Butter of Deep Learning

Fundamentally the following.



Bread & Butter of Deep Learning

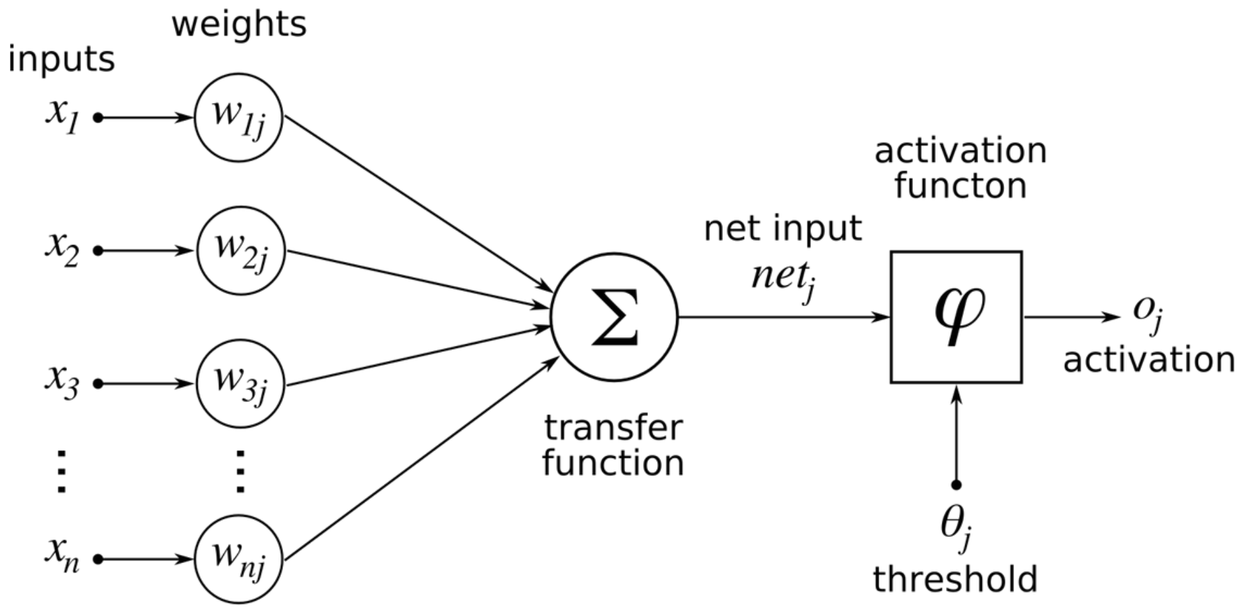
Fundamentally the following

NEURAL NETWORKS

Artificial neural networks (ANN) or connectionist systems are computing systems vaguely inspired by the biological neural networks that constitute animal brains.

BACKPROPAGATION

Backpropagation is a method used to calculate a gradient that is needed in the calculation of the weights to be used in the network.
Backpropagation is shorthand for "the backward propagation of errors"



Repeat this for many times, and you will have Deep Learning.

*From wikipedia

Types of Learning Setting

SUPERVISED LEARNING

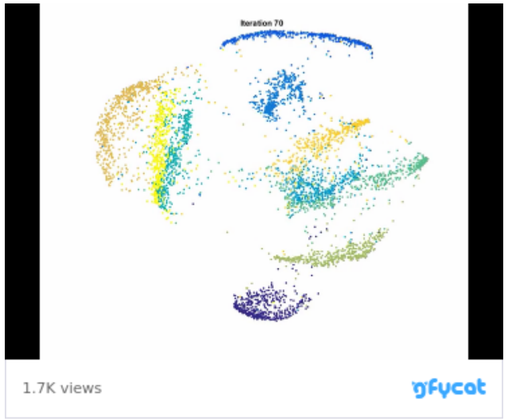
| Self-driving Cars
| Generative Adversarial Nets
| Speech Recognition
| Virtual Assistants (Alexa, Siri)

Most current applications fall under this category

UNSUPERVISED LEARNING

| Fraud Detection in Transactions
| Customer Satisfaction Analysis
| Complex Data Analytics

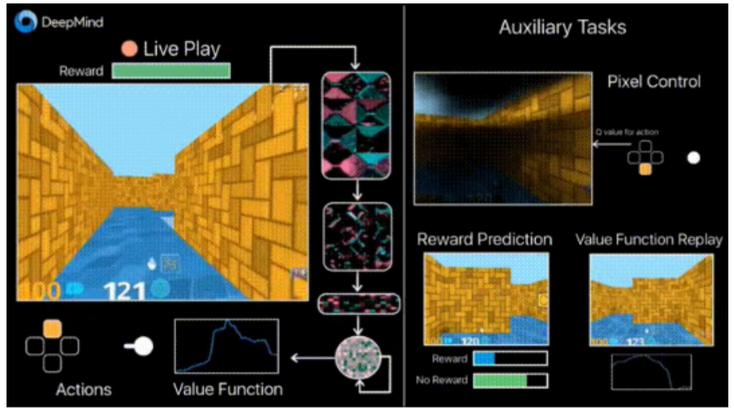
Least heard of in public, but has wide ranging applications in many industries



REINFORCEMENT LEARNING

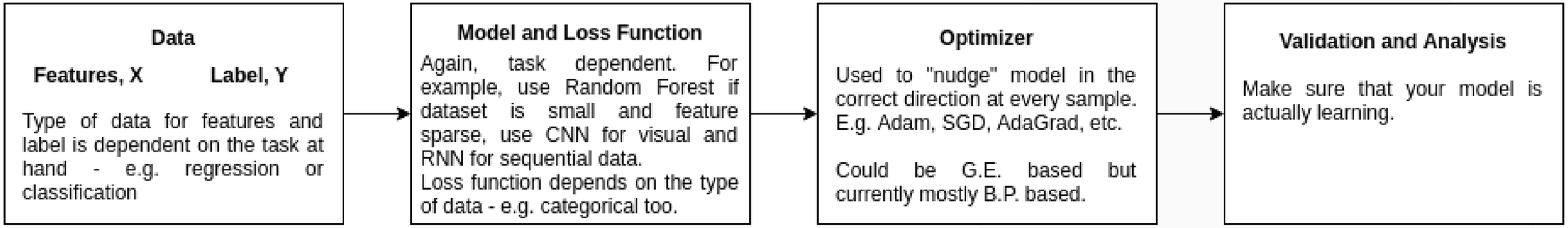
| AlphaGo
| OpenAI Five
| Advance Robotics & Locomotion
| Protein Folding (AlphaFold)

The cool stuff.



Overview of Supervised Learning

Note that the following has been simplified.



For 1stDayHack, we will only deal with Supervised Learning Computer Vision models.

Primary Areas of Deep Learning

COMPUTER VISION

Enabling artificial agents to perceive the world visually in a meaningful manner.

THEORETICAL DL

Studies the fundamental underpinnings and mechanisms underlying Deep Learning. Abstract.

NATURAL LANGUAGE PROCESSING

Deals with all things linguistic, communicative and vocal to an extent.

EFFICIENT IMPLEMENTATION

Deals with the implementation of models and agents in an efficient and large scale manner.

NEURAL ARCHITECTURE SEARCH

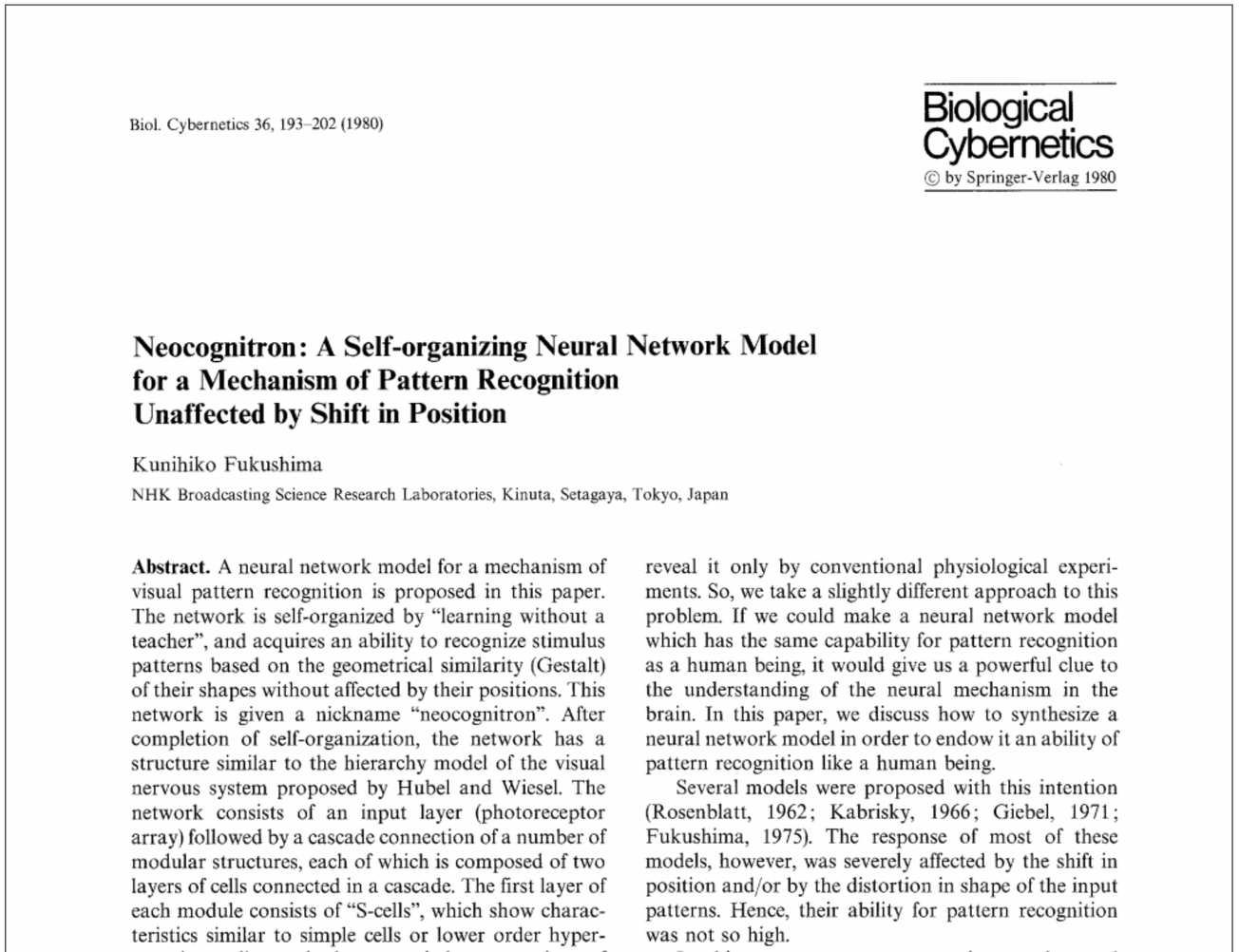
Deals with intelligent designs of Neural Nets.

GENERATIVE NETS

Deals with Generative type networks such as GAN, in their applications, and theoretical studies.

Primary Areas of Deep Learning

Computer Vision



Biol. Cybernetics 36, 193–202 (1980)

Biological Cybernetics

Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position

Kunihiko Fukushima

NHK Broadcasting Science Research Laboratories, Kinuta, Setagaya, Tokyo, Japan

Abstract. A neural network model for a mechanism of visual pattern recognition is proposed in this paper. The network is self-organized by "learning without a teacher", and acquires an ability to recognize stimulus patterns based on the geometrical similarity (Gestalt) of their shapes without affected by their positions. This network is given a nickname "neocognitron". After completion of self-organization, the network has a structure similar to the hierarchy model of the visual nervous system proposed by Hubel and Wiesel. The network consists of an input layer (photoreceptor array) followed by a cascade connection of a number of modular structures, each of which is composed of two layers of cells connected in a cascade. The first layer of each module consists of "S-cells", which show characteristics similar to simple cells or lower order hyper-

reveal it only by conventional physiological experiments. So, we take a slightly different approach to this problem. If we could make a neural network model which has the same capability for pattern recognition as a human being, it would give us a powerful clue to the understanding of the neural mechanism in the brain. In this paper, we discuss how to synthesize a neural network model in order to endow it an ability of pattern recognition like a human being.

Several models were proposed with this intention (Rosenblatt, 1962; Kabrisky, 1966; Giebel, 1971; Fukushima, 1975). The response of most of these models, however, was severely affected by the shift in position and/or by the distortion in shape of the input patterns. Hence, their ability for pattern recognition was not so high.

Early works by Fukushima on the *Neocognitron* (1980). Inspired primarily by the biological brain-vision system.

Later on advanced further by Yann Lecun (current Chief AI Scientist @FAIR) with a convolutional version, Neural Network based model dubbed the *Convolutional Neural Network*.

Modern computer vision models are now mostly convolutional based, working on the foundations laid by these two.

Primary Areas of Deep Learning

Computer Vision

ImageNet Challenge

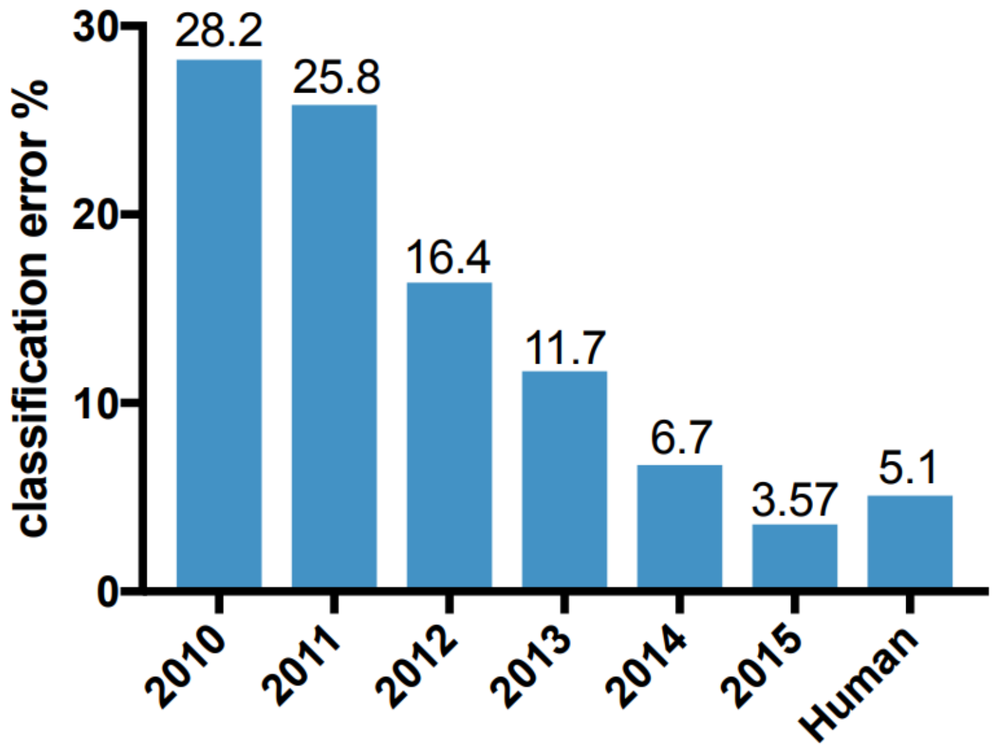
Classification task: produce a list of object categories present in image. 1000 categories.
"Top 5 error": rate at which the model does not output correct label in top 5 predictions

Other tasks include:
single-object localization, object detection from video/image, scene classification, scene parsing

Primary Areas of Deep Learning

Computer Vision

ImageNet Challenge: Classification Task



2012: AlexNet. First CNN to win.
- 8 layers, 61 million parameters

2013: ZFNet
- 8 layers, more filters

2014: VGG
- 19 layers

2014: GoogLeNet
- "Inception" modules
- 22 layers, 5million parameters

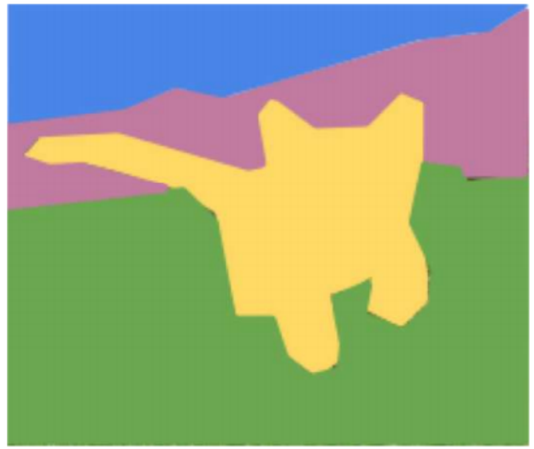
2015: ResNet
- 152 layers

Primary Areas of Deep Learning

Computer Vision

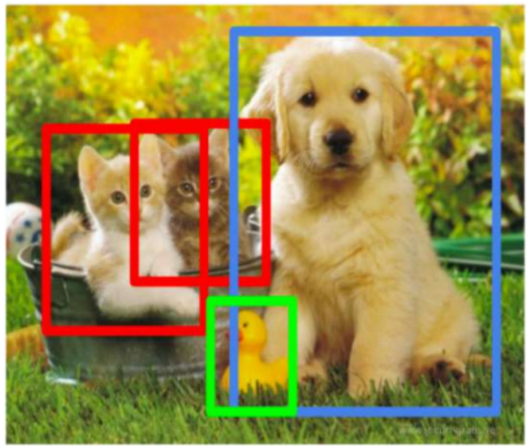
Beyond Classification

Semantic Segmentation



CAT

Object Detection



CAT, DOG, DUCK

Image Captioning

The cat is in the grass.

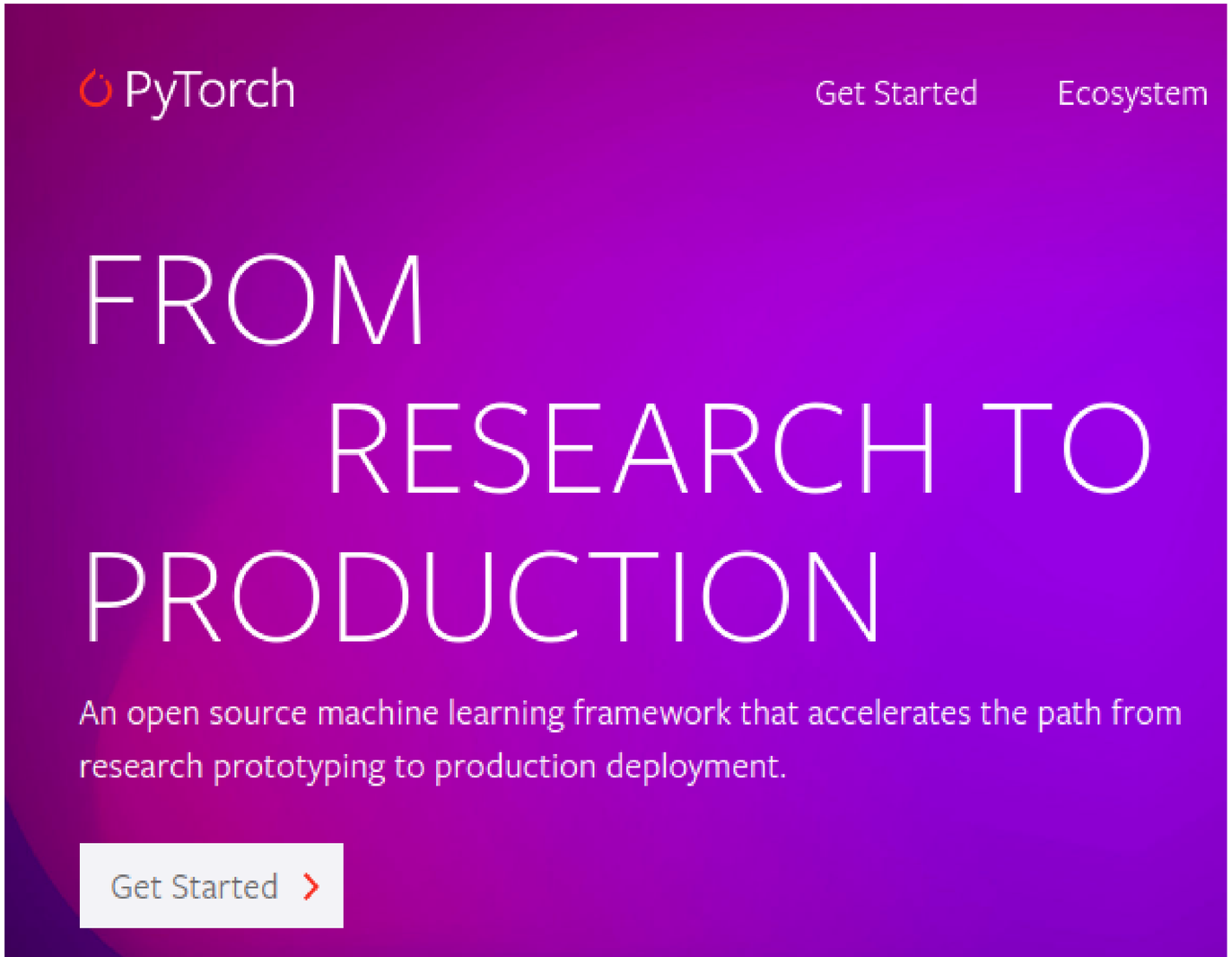
Primary Areas of Deep Learning

Computer Vision

https://playground.tensorflow.org

http://scs.ryerson.ca/~aharley/vis/conv/

PyTorch for Deep Learning Dev



PyTorch - A Python-based Deep Learning library and framework developed by Facebook AI Research (FAIR) and released in 2016. It is built upon the Lua-Based Torch Deep Learning library.

We are using PyTorch because it is easy to pick up and written in such a way that it is tinker-friendly (non-static graph structure). It also has a lot of open-source work and libs that we can leverage on.

Though the new Tensorflow 2.0 has a lot of the above advantages as well.

Demonstration & Code Walkthrough

(Move and Continue on Part 2 Notebook)

Material adopted from:
https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html#sphx-glr-beginner-blitz-cifar10-tutorial-py

Part 3: 1stDayHack Mechanics and 1stDayKit (FDK)

How will the hackathon work and what is FDK?

What is 1stDayHack about?

Practical Machine Learning.
To learn from a problem-solving perspective, not solution-learning.

We are not aiming to show them a bunch of solutions and teach them how these solutions work in theory nor how they are implemented in code.

Instead, we want them to **arrive at the solution based on the problem** at hand, and **how to fit their solution to the problem in a compute-logical manner** (e.g. pipelining).

That is why the idea of **issue-solution** match will be a key part to the judging rubic and why the organizing team have prepared a custom high-level, super-easy to use DL library with several handy state-of-the-art vision tools - **1stDayKit (FDK)** (coming soon!).

1stDayHack Mechanics

The Bucket System.

Thematic Issues Bucket

- Fighting Climate Change
- Modernizing and Digitizing Healthcare
- Revolutionizing the Media and Entertainments
- Improving Formal or Informal Education

Tools Bucket (FDK)

- Object Detector
- Object Classifier
- Face Tracker
- Scene Segmentor
- Pose Estimator
- Deoldifier
- Style Transferer
- Depth Estimator
- Super-Res Generator

In order to facilitate a hackathon regime that would be flexible enough to accommodate the imaginations and desire of the participants while maintaining sufficient fixity to be delineated by a framework within the confines of our tools, mentor knowledge and teaching arrangement, 1stDayHack utilizes a unique Bucket tracking system.

Procedure

1. Select a thematic issue.
2. Specify an exact problem statement.
3. Design a product-solution to answer said problem or challenge statement.
4. Product design and selection of ML-tools from the provided set.

1stDayHack Mechanics

The Bucket System - Example.

Example 1 - Automated Garbage Sorter

1. Select a thematic issue.
    a. *Fighting climate change.*
2. Specify an exact problem statement.
    a. *Recycling is still seldom practised in Malaysia (17.5 % adoption) as our landfills are filling up quick, resources dwindle. People also opt to get rid of their rubbish through open-burning instead of recycling and therefore worsens the already severely compromised air quality of Malaysia.*
    b. *Note that you will need to **back your claims with sufficient and convincing data in your pitch!***
3. Design a product-solution to answer said problem or challenge statement.
    a. *Create a large-scale and scalable automated garbage sorter to make recycling seamless and effortless in order to drive adoption and increase the rate of recycling and proportion of recycled materials.*
4. Product design and selection of ML-tools from the provided set.
    a. *Create a simple front-end to simulate device's interface. Use PyQt or DJango, Bootstrap, etc.*
    b. *Create sorting algorithm-model with FDK. Use object detector or segmentor with an object classifier.*
5. Ready your pitch deck and practice your pitch.

1stDayHack Mechanics

The Bucket System - Example.

Example 2 - AI Interview Coach

1. Select a thematic issue.
    a. *Improving formal or informal education.*
2. Specify an exact problem statement.
    a. *It is increasingly difficulty for youths to secure an employment upon graduation due to a competitive job market and rising number of graduates each year. One problem that stands out and being raised by many employers, is that students do not know how to deliver an interview. Indeed, there is lack of awareness amongst the student population on the importance of delivering a proper interview and how to ace one. How can we solve this?*
    b. *Note that you will need to **back your claims with sufficient and convincing data in your pitch!***
3. Design a product-solution to answer said problem or challenge statement.
    a. *Create a scalable and automated AI interview trainer that will let you refine your ideas, pitch, body language, facial expressions and prosody.*
4. Product design and selection of ML-tools from the provided set.
    a. *Create a simple front-end to simulate device's interface. Use PyQt or DJango, Bootstrap, etc.*
    b. *Create sorting algorithm-model with FDK. Use pose estimator, face tracker and facial classifier. (No audio comp)*
5. Ready your pitch deck and practice your pitch.

1stDayKit (FDK)

Cause alphanumerics are invalid leading variable names.

FDK is a custom Deep Learning Toolkit and high-level API written with Python and based on state-of-the-art PyTorch models.

It aims to make DL as easy as possible, without needing detailed knowledge of ML or DL. It is as simple as

Input + Model + Config Options = Output

Note that there may be some extra steps in between (calling more methods) depending on the requirement of the tool; though it's largely simple.

```
#Example Code
import FirstDayKit import FDK

#Instantiate
toolkit = FDK(options)
detector = toolkit.Detector(options1)
face_tracker = toolkit.FaceTracker(options2)

#Read input and use model
image = PIL.Image.open(path_to_image)
out_detection = detector.detect(image,options3)

#Do something
do_something_of_product(out_detection)
```

1stDayKit (FDK)

Object Detector, Classifier, Pose Estimator, Scene Segmentor

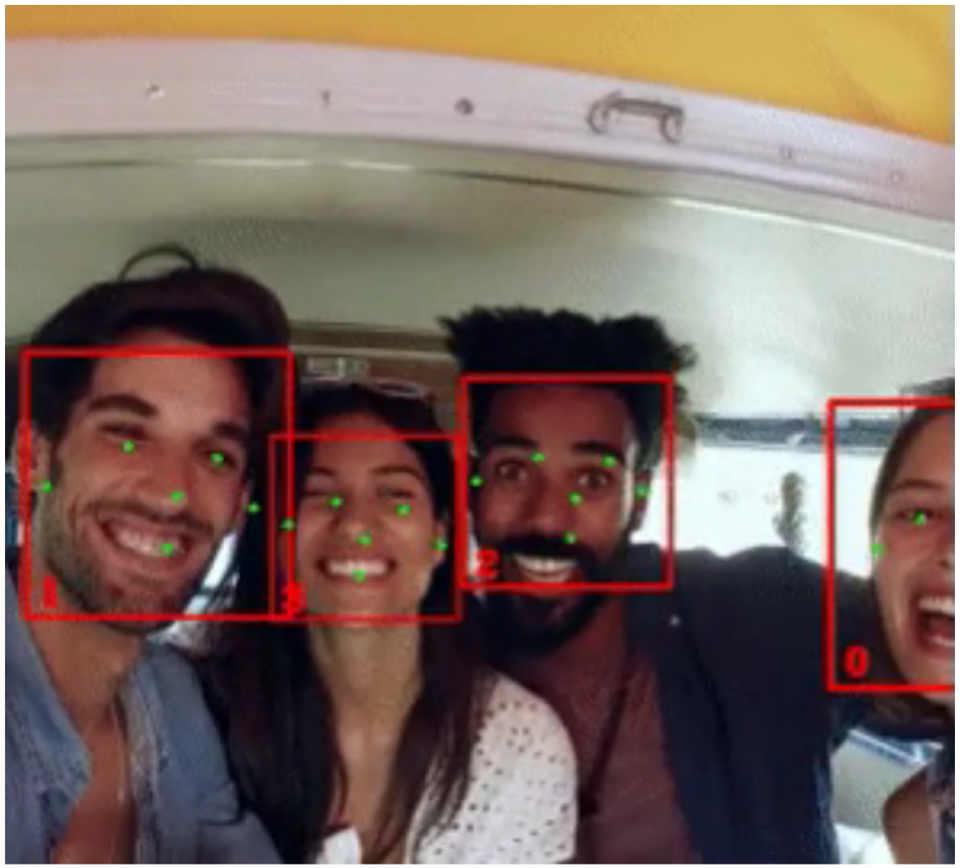
For these tasks, FDK will be using FAIR's Detectron2 package.

Authors:
Yuxin Wu and Alexander Kirillov and Francisco Massa and Wan-Yen Lo and Ross Girshick, 2019.

Source:
https://github.com/facebookresearch/detectron2

1stDayKit (FDK)

Face Tracking



For this tasks, FDK will be using a PyTorch implementation of Google's BlazeFace model.

Authors:
Valentin Bazarevsky, Yury Kartynnik, Andrey Vakunov, Karthik Raveendran, Matthias Grundmann.

Source:
https://github.com/hollance/BlazeFace-PyTorch

1stDayKit (FDK)

Deoldifier

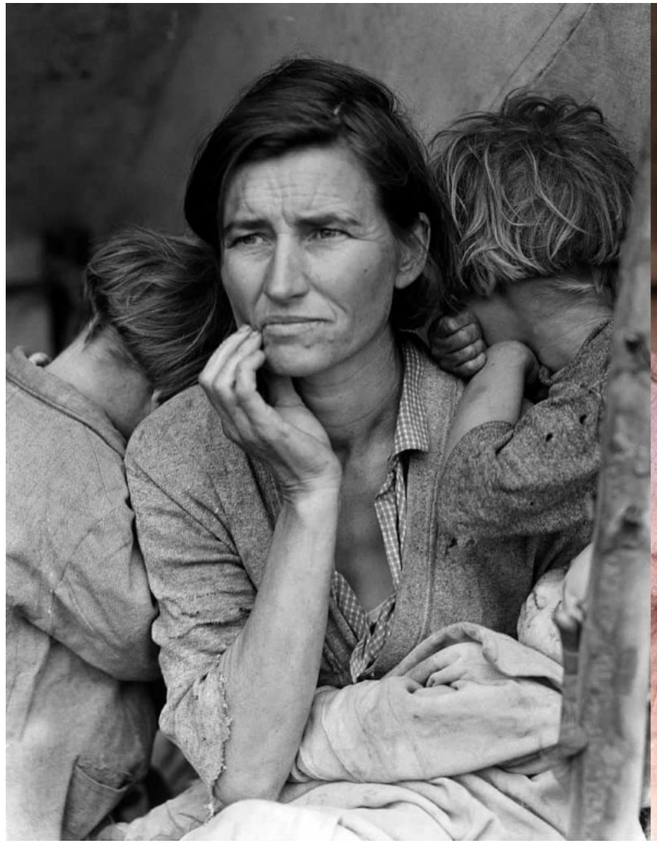
For these tasks, FDK will be using a the awesome Deoldifier lib written by Jason Antic. May be ported separately as it requires FastAI, an additional library.

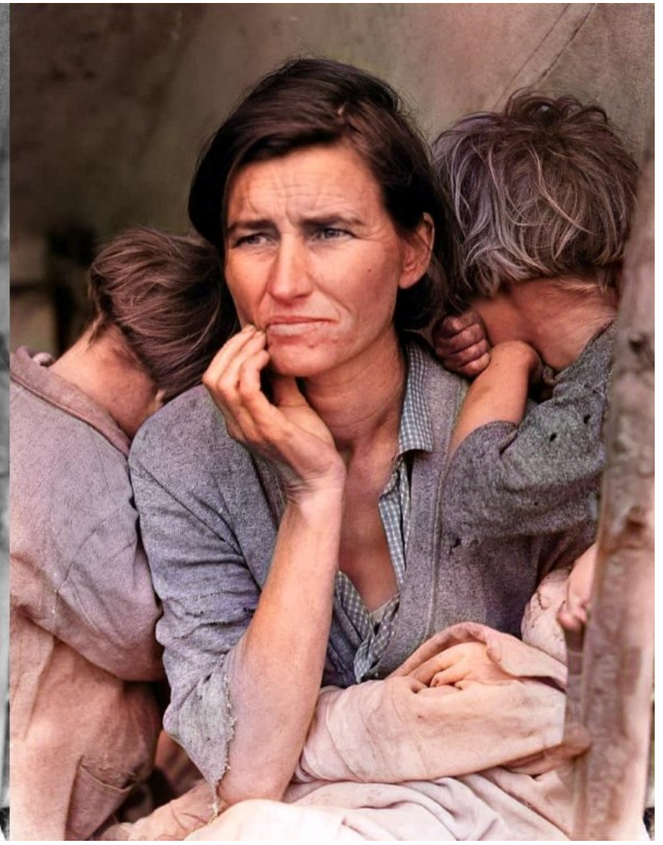
Authors:
Jason Antic.

Source:
https://github.com/jantic/DeOldify

1stDayKit (FDK)

Deoldifier - Super useful in the multi-billion dollar film industry.





1stDayKit (FDK)

Style Transferer. Example notebook available!

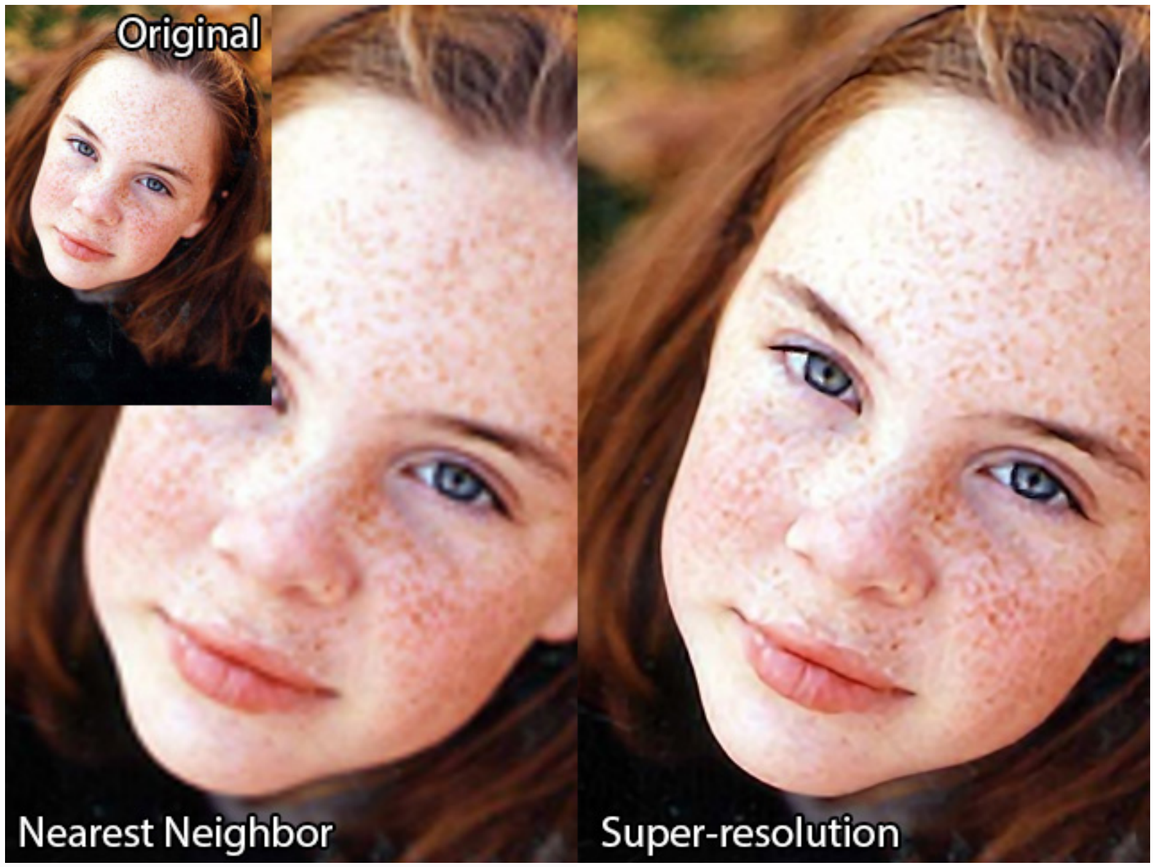
For this, FDK will be using a PyTorch implementation of a style transfer model by Justin Johnson, Alexandre Alahi, Li Fei-Fei from Stanford, implemented by Rusty Mina.

Authors:
Justin Johnson, Alexandre Alahi, Li Fei-Fei.

Source:
https://github.com/rrmina/fast-neural-style-pytorch

1stDayKit (FDK)

Super Resolution Generator



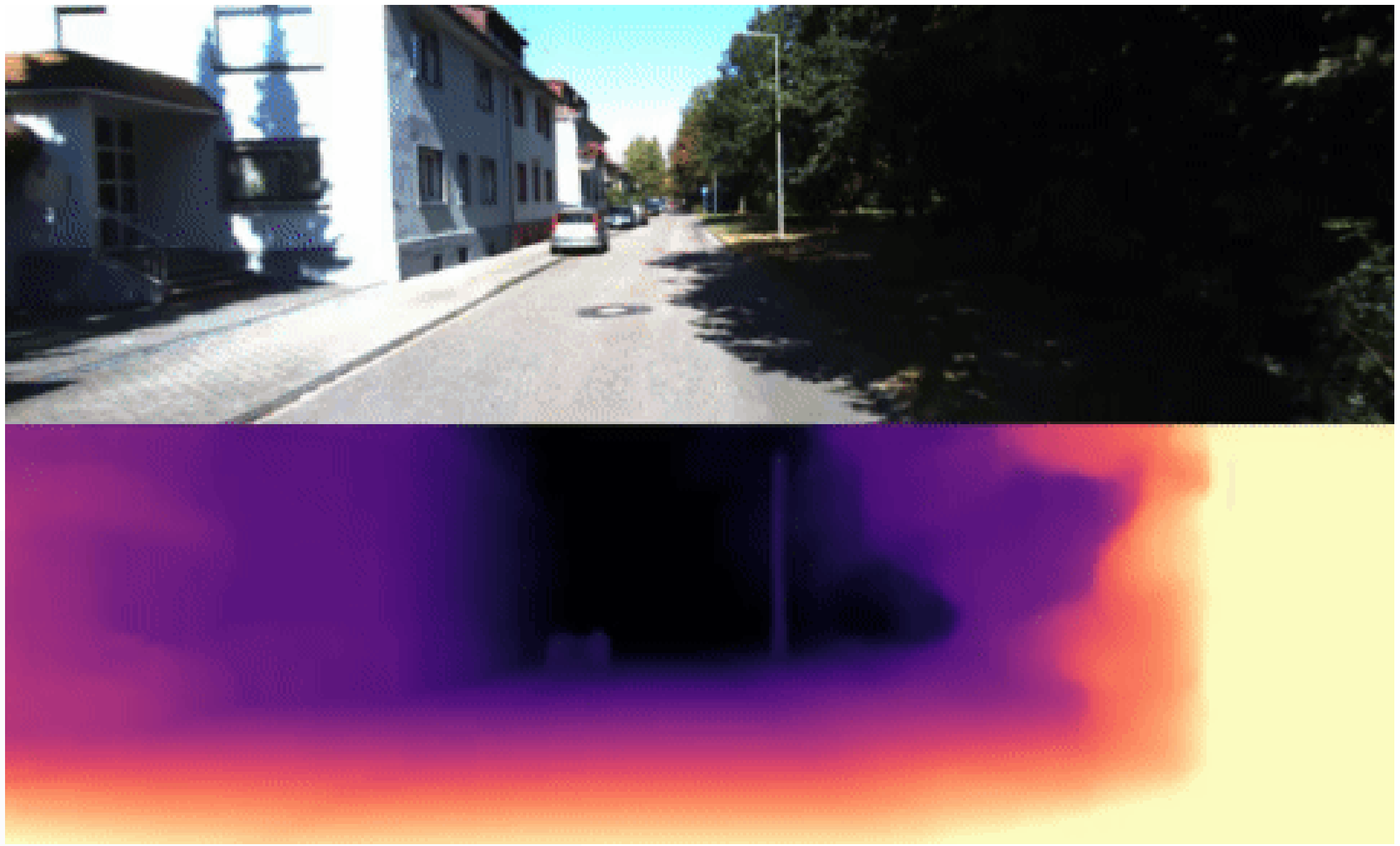
For this, FDK will be using SRFBN.

Authors:
Zhen Li, Jinglei Yang, Zheng Liu, Xiaomin Yang, Gwanggil Jeon, Wei Wu

Source:
https://github.com/Paper99/SRFBN_CVPR19

1stDayKit (FDK)

Depth Estimator



For this, FDK will be using a PyTorch model based on MonoDepth2 writen by Zhang Ze Kuan.

Authors:
Clément Godard, Oisin Mac Aodha, Michael Firman, Gabriel Brostow

Source:
https://github.com/wolverinn/Depth-Estimation-PyTorch

1stDayHack Mechanics

Hackathon rules, format and themes and Mentors.

[Key Info]
No Team: ~10
Pax per Team: 4~6
Age of Demographic: 13 y.o. ~18 y.o. (!)

[Hackathon Requirement]

- Development of a tech MVP. Does not need to be complex, focus on core features. Ideally have a pleasant enough front-end.
- Putting together a compelling pitch deck and pitch. Tell us which thematic issue your team have chosen to tackle, which specific facade of the issue you will tackle, what is the main idea of your solution, and the tech behind your solution. **It should make sense from end-to-end.**

Difficult. That is why we need help from experienced **Mentors and Team Lead**.

1stDayHack Mechanics

Recommendations on Team Strategy for Mentors.

Mentors also acts as Team Lead and requires a lot of hands-on involvement as the participants are still so young and inexperienced. It's your time to shine!

Mentor (Team Lead) to guide their team on selecting issue, identifying a specific problem statement, ideating your solutions, planning and dividing development and overseeing the execution.

For reference, it is advisable that the teams - comprised of 5 to 6 participants and a mentor - be divided into 3 separate tasks. The recommended composition is as follows:

- 2 participants to work on the backend and ML-model of the product-solution.
- 2 participants to work on the frontend of the product-solution.
- 1 or 2 participants to work on the pitch and pitch deck

Tasks for Mentors

Pre-Hackathon Preparations

Get familiar with PyTorch, Python and FDK (when it's available).

Select an issue, develop an idea and game plan in mind.

Based on the idea, design a solution. Figure out what tools you'll need (ML from FDK) and also other auxilliary tools (Bootstrap and Vue for Web App for example).

Draft everything and have a clear picture in your head so you can maximally and effectively direct your team at a snap of finger, and it will also serve as a solid backup plan for you if needed.

How To Get Started?

THEORETICAL FOUNDATION

- Deep Learning AI
- Introduction to Machine Learning
- Stanford CS231n on CNN
- UCL RL Course

IMPLEMENTATIONAL KNOWLEDGE

- Learn Python!
- Pick a Deep Learning library; Tensorflow or PyTorch.

Additional Resources

Useful links

Deep Learning .AI
www.deeplearning.ai/

Coursera's Intro to ML Course
www.coursera.org/learn/machine-learning

Stanford CS231n Computer Vision Course
www.cs231n.stanford.edu/

UCL Course on Reinforcement Learning
www0.cs.ucl.ac.uk/staff/D.Silver/web/Teaching.html

FastAI - Practical AI Course
www.course.fast.ai/

Paper Repository 1
www.distill.pub

Paper Repository 2
www.arxiv.org

Paper Repository 3
www.paperswithcode.com

Deep Learning Textbook - Ian Goodfellow
www.deeplearningbook.org/

Reinforcement Learning Textbook - Sutton & Barto
www.drive.google.com/file/d/1opPSz5AZ_kVa1uWOdOiveNiBFiEOHjkG/view

THANK YOU!

CONTACTS

TUAV LAB

Block E, Ground Floor, Hi-Res Lab
@ Taylor's University

EMAIL ADDRESS

science.endeavor@gmail.com
1stDayHack@gmail.com