Opensource AI/DL Projects for Practitioners

Introduction

This note lists open source github repos, ranging from computer vision and speech recognition to forecast, which can be used as useful tools for the final projects in COTAI AI Foundation courses.

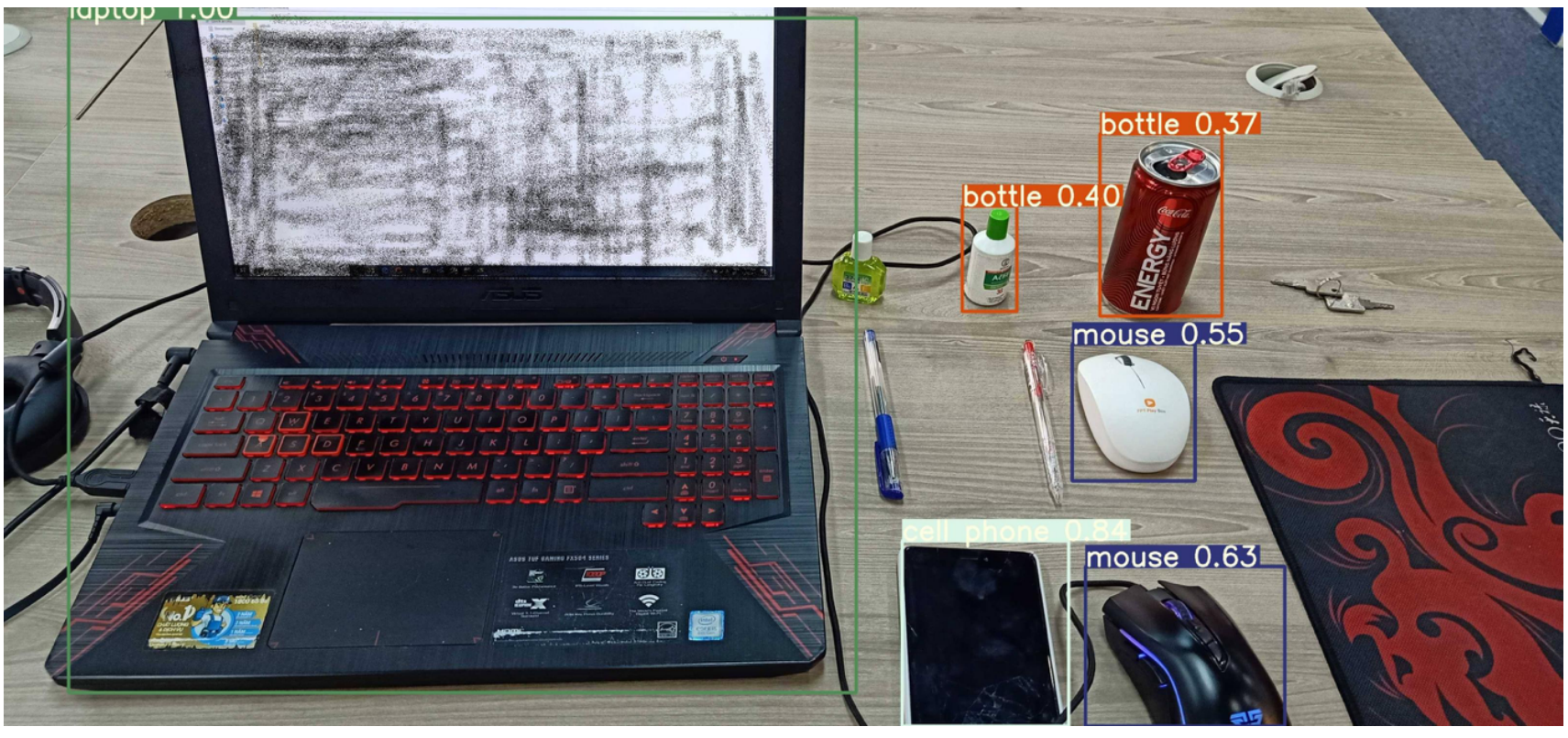
Guideline for final projects

Học viên cần viết bảng tóm tắt cho mỗi project để dễ nắm bắt các ý chính. Dưới đây là 1 ví dụ:

Summary Table

Author	Ultralytics LLC
Title	YOLO v.3 real time object detection
Topics	Ứng dụng trong computer vision, sử dụng thuật toán chính là CNN
Descriptions	Input sẽ là các tấm hình và file .txt có tên tương ứng và chứa 5 thông số của object. đầu tiên là `<class object>` và `<x, y, width, height>` của bounding box chứa vật. khi train xong sẽ trả ra output là file trọng số `weights` . Ta sẽ sử dụng trọng số `weights` đã train để predict bounding box và class của các object trong hình
Links	https://github.com/ultralytics/yolov3 (https://github.com/ultralytics/yolov3)
Framework	PyTorch
Pretrained Models	sử dụng weight đã được train sẵn https://pjreddie.com/media/files/yolov3.weights (https://pjreddie.com/media/files/yolov3.weights)
Datasets	Mô hình được train với bộ dữ liệu cocodataset.org (http://cocodataset.org). Ngoài ra còn có các tập dữ liệu có thể sử dụng: PASCAL VOC, Open Images Dataset V4,...v.v.,
Level of difficulty	Sử dụng nhanh và dễ, có thể train lại với tập dữ liệu khác tốc độ tùy thuộc vào phần cứng và hình ảnh input

Một số hình ảnh huấn luyện bởi giải thuật YOLO



Project suggestion

Time Series Forecast

- **Dự báo lưu lượng server**: cuộc thi trên AIviVN (https://www.aivivn.com/contests/4). Tải Data tại đây (https://www.kaggle.com/nguyenvlm/server-flow-prediction). Bài giải nhóm về nhất (https://forum.machinelearningcoban.com/t/aivivn-5-bandwidth-prediction-2-1st-place-solution/5812)
- **Cuộc thi M4 (2018)** https://github.com/M4Competition

(https://github.com/M4Competition)

Interactive Decision Making

- **Multi-armed bandit (MAB)**: 1 (https://www.kaggle.com/questions-and-answers/87806), 2 (https://www.kaggle.com/sangwookchn/reinforcement-learning-using-scikit-learn).
- **Contextual bandit**: 1 (https://getstream.io/blog/introduction-contextual-bandits/), TensorFlow (https://github.com/tensorflow/models/tree/master/research/deep_contextual_bandits), 3 (https://medium.com/netflix-techblog/ml-platform-meetup-infra-for-contextual-bandits-and-reinforcement-learning-4a90305948ef), 4 (https://striatum.readthedocs.io/en/latest/auto_examples/), 5 (https://www.microsoft.com/en-us/research/project/real-world-reinforcement-learning/), 6 (https://github.com/topics/contextual-bandits), 7 (https://paperswithcode.com/task/multi-armed-bandits).
- **Q-learning**: from scratch (https://medium.com/emergent-future/simple-reinforcement-learning-with-tensorflow-part-0-q-learning-with-tables-and-neural-networks-d195264329d0);

Image Retrieval

- Building a **Similar Images Finder** without any training (VGG pretrained (https://towardsdatascience.com/building-a-similar-images-finder-without-any-training-f69c0db900b5))

Face Recognition

- **Nhận diện mặt người nổi tiếng**: cuộc thi luyện trên AIvìVN (https://www.aivivn.com/contests/7). List of pretrained models (https://forum.machinelearningcoban.com/t/khai-bao-pretrained-models-cho-nhan-dien-nguoi-noi-tieng/4566). Forum mô tả chi tiết các winning solutions (https://forum.machinelearningcoban.com/c/aivivn).
- **Facial embedding for face recognition**: https://github.com/deepinsight/insightface/ (https://github.com/deepinsight/insightface/)
- **More Github repos for face recognition**: https://github.com/topics/face-recognition?o=desc&s=stars (https://github.com/topics/face-recognition?o=desc&s=stars)

NLP

- **Phân loại sắc thái bình luận**: cuộc thi luyện trên AIviVN (https://www.aivivn.com/contests/6).
- **Thêm dấu tiếng Việt**: cuộc thi trên AIviVN (https://www.aivivn.com/contests/3). Winning solution (https://forum.machinelearningcoban.com/t/aivivn-3-vietnamese-tone-

prediction-1st-place-solution/5721/15).

- **Fake News Classification** (WSDM'19). Kaggle contest (https://www.kaggle.com/c/fake-news-pair-classification-challenge/).

Pre-trained model cho các bài NLP

- **Bert for Vietnamese** github repo (https://github.com/lampts/bert4vn) of pretrained models, with ZaloAI'19 winning solution (https://github.com/ngoanpv/zaloqa2019) for Wiki question answering.
- **OpenAI GPT-2** https://talktotransformer.com (https://talktotransformer.com)

Reference

Computer Vision

- **Pyramid pooling module for global context gathering (used in scene parsing)**: https://github.com/Kautenja/keras-pyramid-pooling-module (https://github.com/Kautenja/keras-pyramid-pooling-module)
- **UNet for segmentation**: https://github.com/zhixuhao/unet (https://github.com/zhixuhao/unet)
- **Speech recognition wrapper for Python**: https://pypi.org/project/SpeechRecognition/ (https://pypi.org/project/SpeechRecognition/)
- **Super-resolution GAN for image data augmentation**: https://github.com/tensorlayer/srgan (https://github.com/tensorlayer/srgan)
- **Image colorization**: https://github.com/richzhang/colorization (https://github.com/richzhang/colorization)
- **Detect-and-Track: Efficient Pose Estimation in Videos.** The implementation of an algorithm presented in the CVPR18 paper: https://github.com/facebookresearch/DetectAndTrack (https://github.com/facebookresearch/DetectAndTrack)
- **Augmenter.** https://github.com/mdbloice/Augmentor (https://github.com/mdbloice/Augmentor)
- **Deep Learning: Zero2All.** https://github.com/hunkim/DeepLearningZeroToAll (https://github.com/hunkim/DeepLearningZeroToAll)
- **Anomaly detection in video, CVPR'18.** https://github.com/WaqasSultani/AnomalyDetectionCVPR2018 (https://github.com/WaqasSultani/AnomalyDetectionCVPR2018)

- **EfficientNet for flowers** https://colab.research.google.com/drive/1JmmS_mhSITi5quyvFzmFZvsblMHOp5ll (https://colab.research.google.com/drive/1JmmS_mhSITi5quyvFzmFZvsblMHOp5ll)
- **Skin diseases with data from Udacity** Github repo (https://github.com/udacity/dermatologist-ai).Check also
    1. Github repos for topic https://github.com/topics/skin-cancer (https://github.com/topics/skin-cancer)
    2. paper from Nature'17 by Stanford Uni https://cs.stanford.edu/people/esteva/nature/ (https://cs.stanford.edu/people/esteva/nature/)
    3. Intel mobile solution https://software.intel.com/en-us/articles/doctor-hazel-a-real-time-ai-device-for-skin-cancer-detection (https://software.intel.com/en-us/articles/doctor-hazel-a-real-time-ai-device-for-skin-cancer-detection)
    4. ArXiV survey 1911.11872
- **AI for VR copy & paste** repo (https://github.com/cyrildiagne/ar-cutpaste) & video (https://twitter.com/cyrildiagne/status/1256916982764646402)

Image classifications

- Keras applications: https://keras.io/applications/ (https://keras.io/applications/)
- Other models: https://github.com/qubvel/classification_models (https://github.com/qubvel/classification_models)
- Traffic sign recognition (https://github.com/search?q=traffic+sign).

Object detection

- RetinaNet (https://github.com/fizyr/keras-retinanet). Other resources: 1 (https://github.com/fizyr/keras-retinanet), 2 (https://towardsdatascience.com/review-retinanet-focal-loss-object-detection-38fba6afabe4), 3 (https://vn-zoom.org/threads/tu-viet-cong-cu-nhan-dang-hinh-anh-don-gian-bang-python.11929).
- **YOLOv3**: https://github.com/qqwweee/keras-yolo3 (https://github.com/qqwweee/keras-yolo3)
- **Vehicle detection with YOLO pretrain**: https://github.com/thangnch/yolo_beginner (https://github.com/thangnch/yolo_beginner)
- A Computer Vision based **traffic signal violation detection (https://github.com/anmspro/Traffic-Signal-Violation-Detection-System)** from video footage using YOLOv3 & Tkinter. (GUI Included)
- **THTrieu's Darkflow** https://github.com/thtrieu/darkflow/

(https://github.com/thtrieu/darkflow/)

- **AlexBey's Darknet YOLO v.3** https://github.com/AlexeyAB/darknet (https://github.com/AlexeyAB/darknet)
- **Ultralytics YOLO v.3** https://github.com/ultralytics/yolov3 (https://github.com/ultralytics/yolov3)

Semantic segmentation

- Deeplab: https://github.com/tensorflow/models/tree/master/research/deeplab (https://github.com/tensorflow/models/tree/master/research/deeplab)
- Fast SCNN: https://github.com/Tramac/Fast-SCNN-pytorch (https://github.com/Tramac/Fast-SCNN-pytorch)
- U-Net: https://github.com/zhixuhao/unet (https://github.com/zhixuhao/unet)
- Other models: https://github.com/qubvel/segmentation_models (https://github.com/qubvel/segmentation_models)

Face-related

- InsightFace: https://github.com/deepinsight/insightface (https://github.com/deepinsight/insightface)
- Facial-recognition: https://github.com/ageitgey/face_recognition (https://github.com/ageitgey/face_recognition)
- Face landmarks: https://github.com/1adrianb/2D-and-3D-face-alignment (https://github.com/1adrianb/2D-and-3D-face-alignment)
- Face orientation: https://github.com/shamangary/FSA-Net (https://github.com/shamangary/FSA-Net)

OpenPose

- https://colab.research.google.com/github/tugstugi/dl-colab-notebooks/blob/master/notebooks/OpenPose.ipynb (https://colab.research.google.com/github/tugstugi/dl-colab-notebooks/blob/master/notebooks/OpenPose.ipynb)
  https://deeplearning.vn/post/openpose (https://deeplearning.vn/post/openpose)

- Many colabs:
  https://github.com/tugstugi/dl-colab-notebooks (https://github.com/tugstugi/dl-colab-notebooks)

Real Estate Chatbot System from the repo

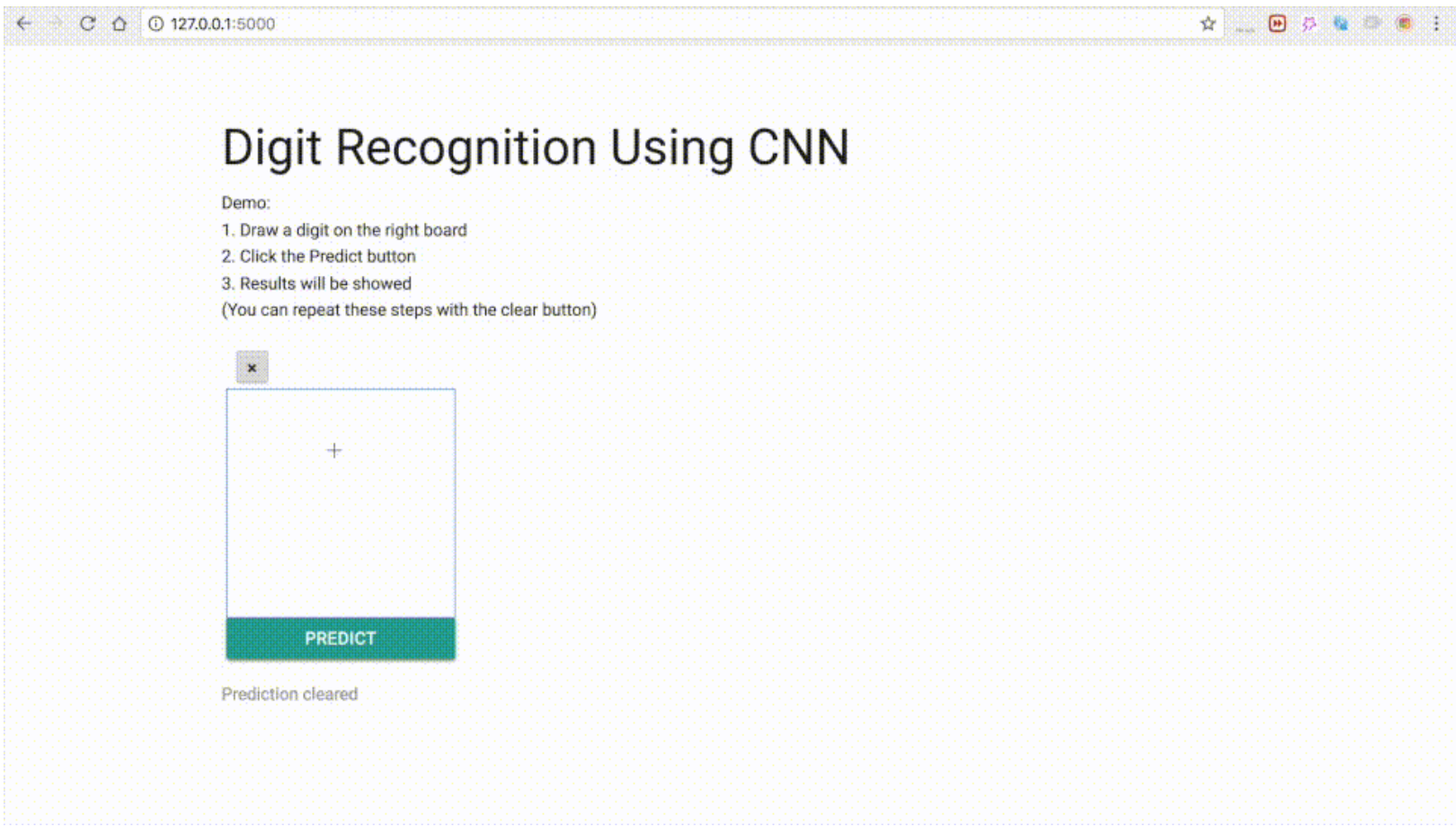
- https://github.com/trungtrinh44/real-estate-chatbot (https://github.com/trungtrinh44/real-estate-chatbot)
  Botkit Anywhere (https://github.com/howdyai/botkit-starter-web), a starter kit for buiding a chatbot using Botkit (https://github.com/howdyai/botkit).
- Check out if it's good: https://github.com/laxmimerit?tab=repositories (https://github.com/laxmimerit?tab=repositories)

Full-Stack Projects:

- **Twitter sentiment analysis** https://towardsdatascience.com/creating-the-twitter-sentiment-analysis-program-in-python-with-naive-bayes-classification-672e5589a7ed (https://towardsdatascience.com/creating-the-twitter-sentiment-analysis-program-in-python-with-naive-bayes-classification-672e5589a7ed)

- **Skin diseases app and web** https://github.com/Waffleboy/HacknRoll2017 (https://github.com/Waffleboy/HacknRoll2017) See also this tutorial https://towardsdatascience.com/classifying-skin-lesions-with-convolutional-neural-networks-fc1302c60d54 (https://towardsdatascience.com/classifying-skin-lesions-with-convolutional-neural-networks-fc1302c60d54)

- **Coronavirus** real time tracking system (https://github.com/hysios/coronavirus) based on Scrapy + influxdb + grafana + NLTK + Stanford CoreNLP.
- MNIST recognition on website (https://github.com/gary30404/convolutional-neural-network-from-scratch-python)



Interactive Demos

RL:
http://projects.rajivshah.com/rldemo/?fbclid=IwAR0JOeOtf62ctC93SIhzBN_0TnpP-QL48bPK-3E0M3YoiaOchGpGoNR6YKk (http://projects.rajivshah.com/rldemo/?fbclid=IwAR0JOeOtf62ctC93SIhzBN_0TnpP-QL48bPK-3E0M3YoiaOchGpGoNR6YKk)
https://www.sicara.ai/connect-4-artificial-intelligence?fbclid=IwAR02F3zh_SqYTDNVHQzH2-Ap9rdzrbrRQBGejLJrUeU4PEOWcT85Y62_4T8 (https://www.sicara.ai/connect-4-artificial-intelligence?fbclid=IwAR02F3zh_SqYTDNVHQzH2-Ap9rdzrbrRQBGejLJrUeU4PEOWcT85Y62_4T8)
https://cs.stanford.edu/people/karpathy/reinforcejs/gridworld_dp.html?fbclid=IwAR2VMOlAJV4N39JCMZQI4PHIYmLO29ij4MRM_RAtHbBk5M2Xb3gvCmU3XGY (https://cs.stanford.edu/people/karpathy/reinforcejs/gridworld_dp.html?fbclid=IwAR2VMOlAJV4N39JCMZQI4PHIYmLO29ij4MRM_RAtHbBk5M2Xb3gvCmU3XGY)

NLP:
https://demo.allennlp.org/named-entity-recognition (https://demo.allennlp.org/named-entity-recognition)
http://text-processing.com/demo/ (http://text-processing.com/demo/)
https://natural-language-understanding-demo.ng.bluemix.net/ (https://natural-language-understanding-demo.ng.bluemix.net/)
https://explosion.ai/demos/ (https://explosion.ai/demos/)
http://dte-nlu-demo.mybluemix.net/self-service/home (http://dte-nlu-demo.mybluemix.net/self-service/home)
https://dash-gallery.plotly.host/dash-nlp/ (https://dash-gallery.plotly.host/dash-nlp/)

CV:
https://poloclub.github.io/cnn-explainer (https://poloclub.github.io/cnn-explainer)
https://www.kairos.com/demos (https://www.kairos.com/demos)
https://skybiometry.com/demo/face-detect/ (https://skybiometry.com/demo/face-detect/)
https://skybiometry.com/demo/face-recognition-demo/ (https://skybiometry.com/demo/face-recognition-demo/)
https://quickdraw.withgoogle.com/ (https://quickdraw.withgoogle.com/)
https://mnist-demo.herokuapp.com/ (https://mnist-demo.herokuapp.com/)
https://cs.stanford.edu/people/karpathy/convnetjs/demo/mnist.html (https://cs.stanford.edu/people/karpathy/convnetjs/demo/mnist.html)
https://dash-gallery.plotly.host/dash-object-detection/ (https://dash-gallery.plotly.host/dash-object-detection/)
https://www.nvidia.com/en-us/research/ai-playground/ (https://www.nvidia.com/en-us/research/ai-playground/)
http://ganpaint.io/demo/?project=church (http://ganpaint.io/demo/?project=church)

http://www.cs.toronto.edu/~graves/handwriting.html (http://www.cs.toronto.edu/~graves/handwriting.html)
http://horatio.cs.nyu.edu/ (http://horatio.cs.nyu.edu/)

ML:
https://dash-gallery.plotly.host/dash-svm/ (https://dash-gallery.plotly.host/dash-svm/)
https://dash-gallery.plotly.host/dash-live-model-training/ (https://dash-gallery.plotly.host/dash-live-model-training/)
https://dash-gallery.plotly.host/dash-tsne/ (https://dash-gallery.plotly.host/dash-tsne/)
https://dash-gallery.plotly.host/dash-mnist-explorer/ (https://dash-gallery.plotly.host/dash-mnist-explorer/)
https://dash-gallery.plotly.host/dash-spatial-clustering/ (https://dash-gallery.plotly.host/dash-spatial-clustering/)
https://playground.tensorflow.org/ (https://playground.tensorflow.org/)

Other Resources

Tiếng Việt

Các bài viết tutorial trên Viblo về **ML (https://viblo.asia/tags/machine-learning)**, **Deep Learning (https://viblo.asia/tags/deep-learning)**, **Artificial Intelligence (https://viblo.asia/tags/artificial-intelligence)** (**AI (https://viblo.asia/tags/ai)**).

Một số bài thực hành nhỏ vui bởi **Mì AI (http://ainoodle.tech/)**.

Tiếng Anh

Kaggle (https://www.kaggle.com/): excellent resources (https://towardsdatascience.com/how-to-farm-kaggle-in-the-right-way-b27f781b78da) (competitions (https://www.kaggle.com/competitions), tutorials (https://www.coursera.org/learn/competitive-data-science), kernels (https://www.kaggle.com/kaggle/kaggle-blog-winners-posts), solutions (https://www.kaggle.com/rajiao/winning-solutions-of-kaggle-competitions)).

Datasets: Kaggle datasets (https://www.kaggle.com/datasets), Google dataset search (https://toolbox.google.com/datasetsearch).

LeadingIndia.AI (http://LeadingIndia.AI) Internship Projects with Code (https://www.leadingindia.ai/internshipproject)

Public IP Cameras in Vietnam (http://www.insecam.org/en/bycountry/VN/?page=1) Nhấn F12 tìm cái link của cái video đang stream rồi tìm cách viết code listen video qua cái link đó.

Hướng dẫn làm đồ án tốt nghiệp (final project)

1. Học viên làm đồ án tốt nghiệp theo nhóm hoặc cá nhân.
2. Học viên có thể chọn đồ án (project) ưa thích bên ngoài, hoặc chọn 1 trong các gợi ý ở mục Reference (https://hackmd.io/pVC4gaZXQaSI8BJQ3g3dUA?both#Reference)
3. Sau khi đã chọn xong, học viên nên dành thời gian viết bản tóm tắt đồ án theo mẫu (https://hackmd.io/pVC4gaZXQaSI8BJQ3g3dUA?both#Summary-Table). Đây là các đề mục do đội ngũ giảng viên gợi ý nhằm giúp học viên biết cách nhanh chóng nắm bắt các ý chính để bắt đầu dự án nhanh gọn.
4. Trong ngày báo cáo đồ án, học viên thuyết trình sản phẩm, chạy trên colab hoặc laptop hoặc server hoặc mobile.
5. Ngoài ra, học viên còn cần phải hoàn thành bài test (làm trên google form) trong buổi báo cáo đồ án.
6. **Trước** ngày báo cáo đồ án, học viên cần:
   - Nộp code bằng `github` hoặc `gitlab` trên slack
   - File README trong code tối thiểu cần có bản tóm tắt đồ án theo mẫu (https://hackmd.io/pVC4gaZXQaSI8BJQ3g3dUA?both#Summary-Table), kèm theo hình (hoặc video) ghi lại kết quả sản phẩm, kèm theo poster cung cấp thông tin chi tiết theo mẫu dưới đây:

Template for Final Project poster

Title	**Your project title**
Team	Include your names and student emails, (yourname, yourfriendsname, vangough}@stanford.edu
Predicting	Briefly explain the motivation for your topic, what you built, and the results. It's easier to think of this as a quick summary of the inputs and outputs. **(5 sentences max)**
Data	Exactly where did your data come from and what does your contain? (ie. What are in the rows and columns? Are examples labeled with ground truth? If you have images, are they color, normalized, etc?) **(2-3 sentences max)**
Features	How many features do you have and which features are the raw input data (ex. color, weight, location, etc) vs. features you have derived (ex. ICA, Gaussian Kernel)? Why they are appropriate for this task? **(3-4 sentences max)**
Models	Exactly which model(s) are you using? Write out the basic math formulas and clearly note any modifications or additions. If you have more than one model, make subsections for each. **(3-4 sentences max)**
Results	Make a compact table of results. Each row should be a different model. The columns should be the training error and the test error. List how many samples are in each of the training and testing data sets. Obviously, these sets should be different. **(1-2 sentences max + 1 table max)**
Discussion	This is where you share your thoughts about your project. (Hopefully you have a few interesting interpretations!) Briefly summarized what just happened. Briefly explain whether or not you expected your results. If your results were good, explain why. If they were not good, explain why. **(6 sentences max)**
Future	If you had another 6 months to work on this, what would you do first? **(2-3 sentences max)**
References	IEEE style (https://ctan.org/topic/bibtex-sty?lang=en) is fine

Thời lượng thuyết trình: 10 -15p (chưa bao gồm thời gian hỏi và trả lời của các bạn khác và giảng viên của lớp)
Bố cục (đề xuất):

- Giới thiệu: TEPFA của bài toán
  - Task:
  - Experience: data, lấy ở đâu, chất lượng như nào, số lượng ra sao, data có nhiễu hay lỗi không
  - Performance: hàm loss function, metric (e.g. accuracy)
  - Function: trình bày pretrained model nào (hoặc lấy model ở đâu để train)
    - Preferable: trình bày một chút về model detail
  - Algorithm: nói qua về gradient descent
- Body:
  - Baseline là gì, độ chính xác như nào
    - Preferable: tìm hiểu về baseline mình dùng và mô tả ngắn gọn mô hình đó
      - Source: tutorials trên mạng
  - Đối với các nhóm chỉ có baseline:
    - Chạy thử mô hình
    - Nhìn bằng mắt các dự đoán sai và đoán nguyên nhân (lỗi dữ đoán nào phổ biến)
      - Error analysis
  - Đối với các nhóm đã hoàn thiện:
    - Error analysis (như ở trên)
    - Hướng giải quyết để cải thiện solution
      - Preferable: nói rõ là cải thiện được bao nhiêu % so với solution gốc
- Conclusion:
  - Bài học rút ra
  - Thiên về research: đào sâu về lí thuyết của mô hình dự đoán
  - Thiên về engineering: các hướng mình thấy potential cho việc cải thiện solution
  - Thiên về business: các hướng có thể áp dụng được bài toán của mình