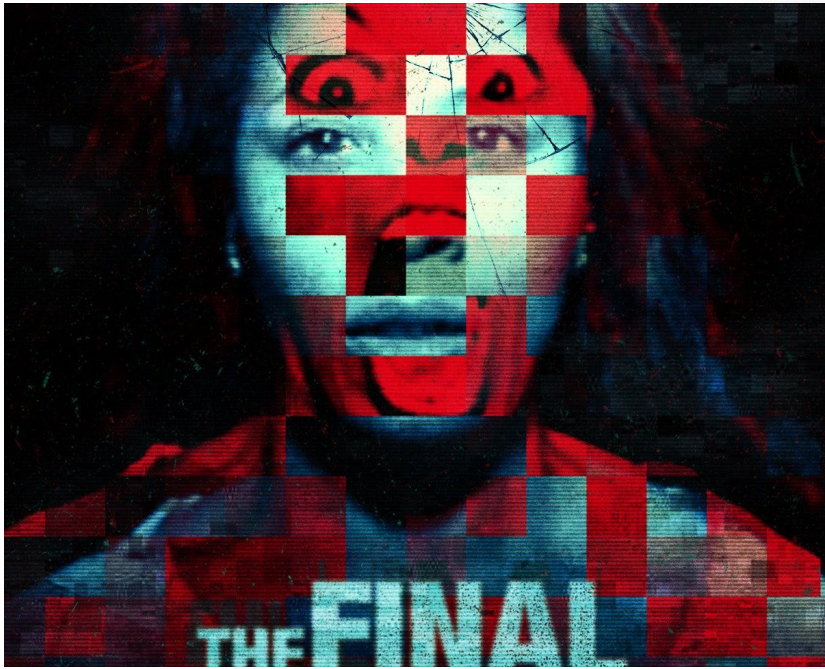
FINAL PROJECT GUIDELINE

Chih-Chung Hsu (許志仲)
Institute of Data Science
National Cheng Kung University
https://cchsu.info



First pick a project

- Critical considerations
  - What data you used
    - DON’T TRY TO COLLECT YOUR OWN DATA.
    - I mean you can, it’s just a lot of effort + time. At your own risk
  - Packages
    - Tensorflow, PyTorch, Keras, MXNet, etc
  - Network Architecture
  - A problem can solve by “learning” ➔ Data-driven
  - Tips:
    - start with focusing most of your effort right now to data
    - Do a little bit of Googling each day

Get some inspiration

- Look up highly publicized material:
  - DeepMind, OpenAI , Google Brain, Facebook FAIR, etc
  - UIUC, CMU, MIT, Stanford's research teams, something like that
- Try to find cool web demos like this: https://worldmodels.github.io/
- Search "awesome {Transformer, GAN, computer vision, NLP} Github"
  - https://github.com/jbhuang0604/awesome-computer-vision
  - Play around and pull repos! Get a feel for the code and how readable it is
- Check the project around you these years!
  - See what works and what doesn't!

Resource you can use

- Papers with code:
  - https://paperswithcode.com/sota
- IEEE 2019 summary report on GANs:
  - https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8667290&tag=1
- CV for autonomous vehicles:
  - https://arxiv.org/pdf/1704.05519.pdf
- A survey on Transformer:
  - https://arxiv.org/abs/2009.06732

Project flavors (not exhaustive)

- Experiment with improving an architecture on a predefined task
- The case study: Apply an architecture to a dataset in the real world
- The challenge: compete in a predefined competition (AICrowd, Kaggle, Codalab)
- The researcher: join a NCKU/company research project
  - Or your own projects
- Stress test or comparison study of already known architectures
- Design your own unit (complex layer, objective function, optimizer, etc)
- Mix and match domains! (e.x use a CV GAN in Transformer)
- Don’t do video (unless you got money and time!!!)

Design think it a lil

- Have each member of your team flesh out 20 quick ideas down on paper before meeting. Don't be afraid to get creative
- Filter out list by doing quick Google searches on data
  - Anything below GB scale of data
    - Won't work or a lot of work
- If you have an idea, Google it first! Don't want to "just" reproduce the same result.
  - There's probably a Github with your project already
- Pay attention to how long and much data the models you see are trained on
- Find pattern in data + architecture combos
- Ask are there little tweaks or other experiments that haven't been done yet?
- Can you extend the idea in one paper with another?
- Which idea gives you more things to experiment with?
- How can you get pretty images / figures?

Paper reading process

- Don’t read all of it
- Look at the figures and captions before anything
- First pass reading order
  - Abstract
  - Methods
  - Results
  - Conclusion
- Plenty of blogs, Github repos, websites that summarize or explain papers even better!

Try to avoid this scenario

- Nothing special in data pipeline. Uses existing packages
- Team starts late. Move now! (by milestone)
- Explore 3 architectures with code that already exists
  - One ResNet, then a SENet, and then SKNet… Nothing big difference
- Only ran models until they got ~65% accuracy
- Didn’t hyperparameter search much (not important for your work, but helps for your competition if you want)
- A few standard graphs: loss curves, accuracy chart, simple architecture graphic
- Conclusion doesn’t have much to say about the task besides that it didn’t work

Aim for this

- Workflow set-up configured ASAP
- Have running code and have baseline model running and fully-trained
- Creative hypothesis is being tested
- Mixing knowledge from different aspects in DL
- Have a meaningful graphic (pretty or info rich)
- Conclusion and Results teach me something
- Optional
  - interactive demo
  - novel / impressive engineering feat
  - good results

Milestone goals

- Give us the preliminary results!! Not just a paper with idea only
- Data source explained correctly
  - Give the true train/test/val split
  - Number training examples
  - Where you got the data
- What Github repo you used
- Ran baseline model have results
  - You should have at least a promising result by milestone
- Data pipeline should be in place
- Brief discussion of initial, preliminary results
- Reasonable literature review (3+ sources)
- 1-2 page progress report. Not super formal