Introduction to Deep Learning: Projects

ECE685D Fall 2020

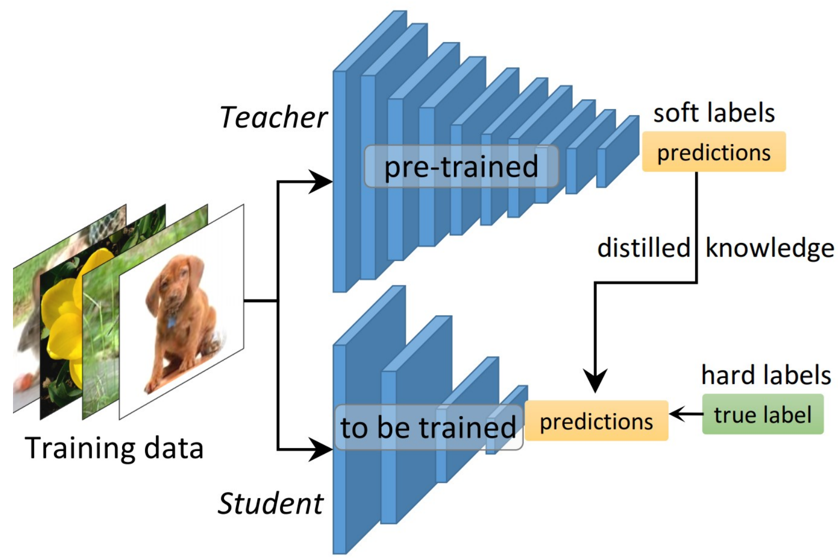
Outline

- Implementation-oriented projects
  - Nikhil (4 projects)
  - Xiaoyu (3 projects)
  - Jinyuan (3 projects)
  - Suya (1 project)
  - Mohammad (5 projects)
  - Marko (4 projects)

- Comprehensive reviews of advanced topics

N1. Knowledge Distillation in Deep Networks (cap: 2)

- Knowledge distillation deals with the problem of training a smaller model from a high capacity source model so as to retain most of its performance.
- Transfer knowledge from a large network (called the “teacher” network) with millions of parameters to a small network (called the “student” network).
- Application: Leads to a compressed model which can be deployed on mobile devices.
- From a scientific standpoint: Using knowledge from a teacher network to train a student network leads to lower generalization error with a wider optima. **Why does this work better than just using the ground truth?**



N1. Knowledge Distillation in Deep Networks

- Goal: Learn a classifier using knowledge distillation in the following two scenarios:
  - The training data of the teacher network is available.
  - The training data of the teacher network is NOT available.

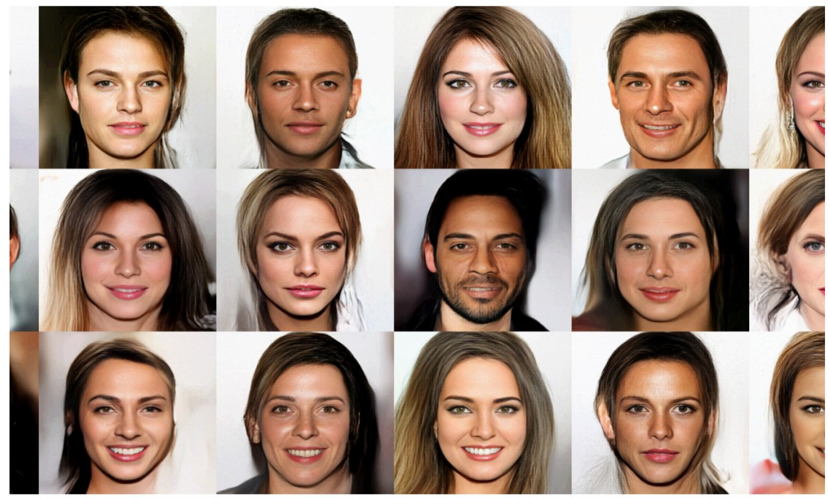
- Datasets: See the references below for datasets.

- References
  - *Distilling the knowledge in a neural network* *(NeurIPS 2014 Workshop)*
  - **Zero-Shot Knowledge Distillation in Deep Networks (ICML 2019)**

N2. Anomaly detection using generative models (cap: 2)

Introduction:

- The discipline of generative modeling has experienced enormous leaps in capabilities in recent years.
- In this project, we will do anomaly detection using **likelihood-based generative methods**.
- Example motivation: Given a generative model trained on human faces, we would like the model to detect an image that is not a human face.
- Keywords: Normalizing Flows, Variational Autoencoder, Autoregressive models.



N2. Anomaly detection using generative models

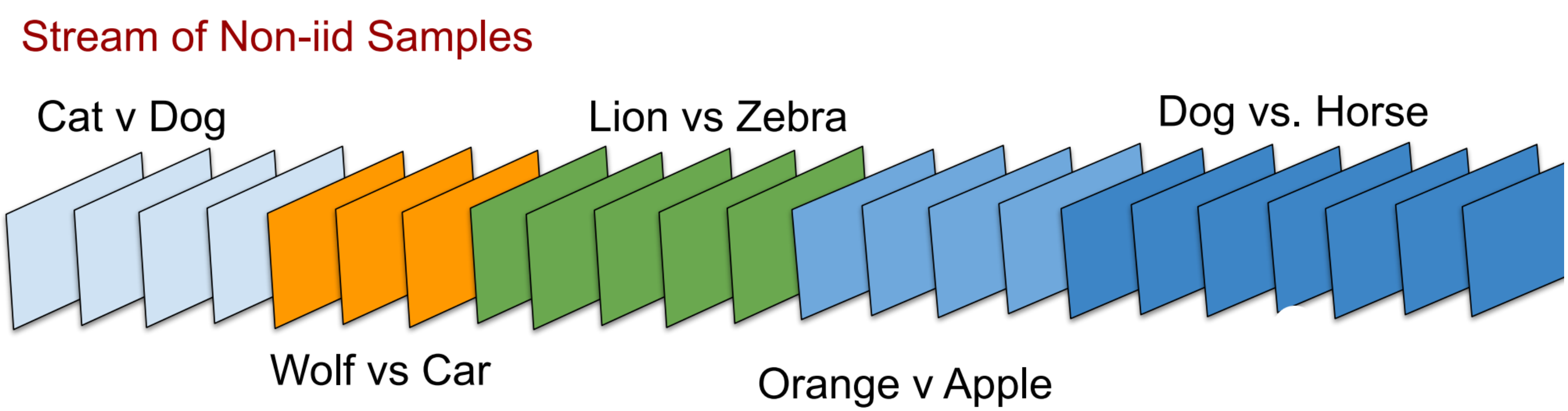
- **Goal**: Learn a generative model that can be used for anomaly detection.
- **Project Outline**:
  - Show empirically that log-likelihood alone is NOT a good estimate for anomaly detection. (See [1] below).
  - Next, propose a technique that can detect anomalies. (Use inspiration from recent work [2,3])

- **Relevant References:**
  - [1] Do Deep Generative Models Know What They Don't Know? (ICLR 2019)
  - [2] WAIC, but Why? Generative Ensembles for Robust Anomaly Detection
  - [3] Unsupervised Out-of-Distribution Detection with Batch Normalization

N3. Life-long Continual Learning (cap: 3)

- **Motivation**: Humans can easily adapt to new tasks by acquiring and accumulating knowledge sequentially. Learning in such an incremental fashion involves preserving knowledge of previously observed tasks.

- In real-world, the data associated with a new task is often arbitrarily different than the data associated with previously observed tasks. Specifically, the sequence of data points associated with different tasks arrive in a non-iid fashion. For e.g. in the image below, we have 5 tasks arriving in an incremental fashion.

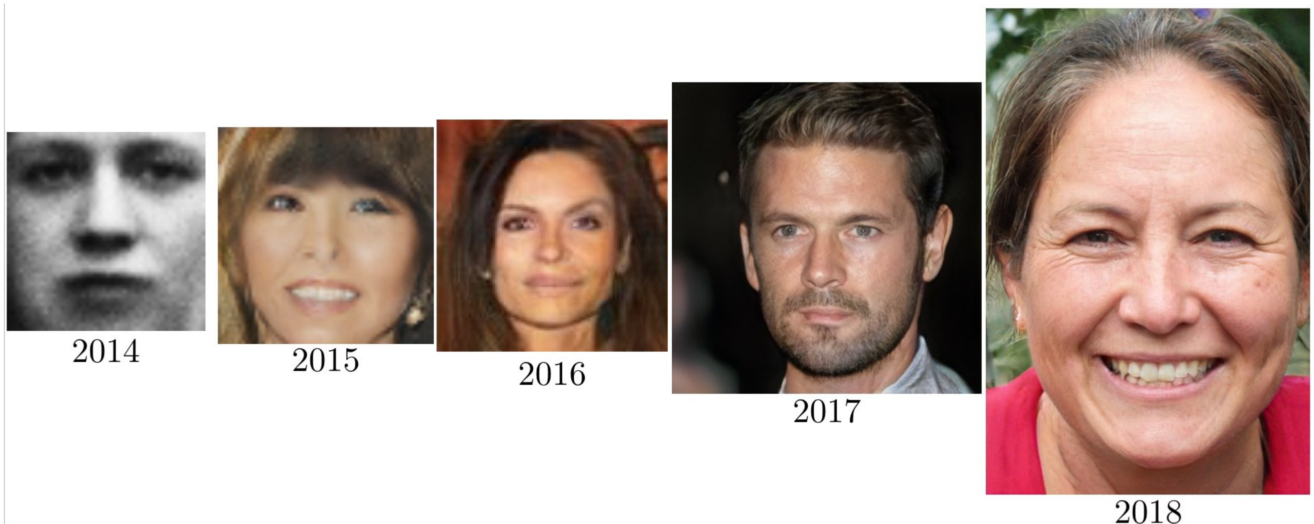


N3. Life-long Continual Learning

- **Goal**: Learn a deep neural network in a continual learning setting, which learns different tasks arriving in an incremental fashion in a non-iid fashion. While learning a new task, the model should not forget previously seen tasks.
- **Project Outline**
  - Review existing methods (3 main categories: regularization, replay, expansion).
  - Implement 2-3 baselines (preferably recent ones) for continual learning methods. (**See references**).
  - Propose improvements that may lead to better average accuracy across all seen tasks.
  - Report average accuracy over all tasks seen on the following three continual learning benchmarks:
    - Split MNIST
    - Permuted MNIST
    - Split CIFAR-10/100
- **References**

[1] Three scenarios for continual learning (NeurIPS Continual Learning workshop, 2018)
[2] Progressive Neural Networks
[3] iCaRL: Incremental Classifier and Representation Learning (CVPR, 2017)
[4] Learning without forgetting (IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017)

N4. Recent Advances in GANs (cap: 2)



- Improvement in GANs over the past years:
  - Generative Adversarial Networks (NeurIPS 2014)
  - Unsupervised learning with DCGAN (ICLR 2016)
  - Coupled GAN (NeurIPS 2016)
  - Progressive GANs (ICLR 2018)

N4. Recent Advances in GANs

- **Goal:** Analyze different regularization techniques that have led to stabilized training in GANs.

- **Project Outline**:
  - Review various regularization techniques used in stabilizing GANs (See references).
  - Implement and compare at least one regularization technique over CIFAR-10/100, SVHN or CelebA dataset, and compare it with vanilla GAN (w/o regularization). Show training curves of discriminator and generator.
  - Report commonly used metrics (e.g. FID and Inception score) to evaluate GAN results.
  - Do a qualitative analysis on the synthetic data generated. For instance, check for mode collapse and interpolate between two points in the latent space, etc.
  - Identify limitations of current methods and propose improvements.

- References:
  - Wasserstein GAN (ICLR 2017)
  - Improved Training of Wasserstein GANs (NeurIPS 2018)
  - Spectral Normalization for GANs (ICLR 2018)
  - **Stable Rank Normalization for Improved Generalization in Neural Networks and GANs** (ICLR 2019)

X1 - Image Steganalysis (cap: 3)

- **Introduction:**
  - Steganography:
    - Conceal secret information within an ordinary content
  - Steganalysis:
    - Detect the secret information hidden with Steganography

X1 - Image Steganalysis

- **Project Goal:**
  - Train a deep neural network to detect information hidden in images

- **Project Outline:**
  - Get familiar with the dataset
  - Research the proper model architecture for the task
  - Implement the network and train
  - Evaluate the performance of the network and improve

- **Reference:**
  - https://www.kaggle.com/c/alaska2-image-steganalysis/data

X2 - COVID19 Global Forecasting (cap: 2)

- **Introduction:**
  - There are many confirmed cases and fatalities of COVID19 around the world every day.
  - Can we use machine learning to forecast the future infection?
  - Given historic data, what can be inferred about the future?

X2 - COVID19 Global Forecasting

- **Project Goal:**
  - ➢ Train deep neural networks to predict the future confirmed cases and fatalities

- **Project Outline:**
  - ➢ Get familiar with the dataset
  - ➢ Explore different models for this task, e.g., CNN, RNN, …
  - ➢ Implement the networks and train
  - ➢ Evaluate the performance of different networks and compare them

- **Reference:**
  - ➢ https://www.kaggle.com/c/covid19-global-forecasting-week-1/data
  - ➢ https://www.kaggle.com/c/covid19-global-forecasting-week-2/data
  - ➢ https://www.kaggle.com/c/covid19-global-forecasting-week-3/data
  - ➢ https://www.kaggle.com/c/covid19-global-forecasting-week-4/data
  - ➢ https://www.kaggle.com/c/covid19-global-forecasting-week-5/data

X3- Detecting Deepfakes (cap: 3)

- **Introduction:**
  - Deepfakes:
    - Synthetic media generated by deep neural networks

  - Can we detect the Deepfakes?

X3 - Detecting Deepfakes

- **Project Goal:**
  - Train a deep neural network to distinguish between real faces and fake faces

- **Project Outline:**
  - Get familiar with the dataset
  - Download and preprocess a subset of the dataset
  - Research for different detection methods and model architectures
    - This is the significant part of this project and you are expected to provide comprehensive overview
  - Choose a method and train a classifier to detect fake faces
  - Evaluate the performance of your network and try to improve

- **Reference:**
  - https://github.com/ondyari/FaceForensics/tree/master/dataset

J1 – 3D Object Classification (cap: 2)

- **Introduction:**
  - ➢ **Deep learning on 3D data:**
    - ➢ **Multiple representation of 3D data**
    - ➢ **Point cloud:**
      - ➢ **Unordered set of vectors**
      - ➢ **A point cloud is represented as a set of 3D points**
      - ➢ **Each point is vector of its (x, y, z) coordinate as well as some other features, e.g., color.**

    - ➢ **Unordered, local structure from neighbor points, invariant to transformation**

J1 – 3D Object Classification

- **Project Goal:**
  - ➢Explore deep learning architectures for the classification of point clouds

- **Project Outline:**
  - ➢Design new methods (model architectures) and compare with previous work.
    - ➢e.g., design new method to achieve order invariance
  - ➢Evaluate the proposed method on benchmark datasets.

- **Reference:**
  - Qi et al. "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation". In CVPR 2017.

J2 – Multi-Label Image Classification (cap 2)

- **Introduction:**
  - ➢ **Real world images generally contains multiple labels**
  - ➢ **Train classifiers for each category and leverage ranking or threshold to obtain prediction results**
    - ➢ **Unable to exploit label dependencies in an image**

J2 – Multi-Label Image Classification

- **Project Goal:**
  - ➢Learn how to design methods for multi-label image classification
- **Project Outline:**
  - ➢Design new methods (model architectures) for multi-label image classification and compare with previous work.
  - ➢Evaluate the proposed method on benchmark datasets.
- **Reference:**
  - Wang et al. "CNN-RNN: A Unified Framework for Multi-label Image Classification". In CVPR, 2016.

J3– Membership Inference Attacks (cap: 3)

- **Introduction:**
  - **Given an input x, the output of machine learning model f is a probability distribution y over the possible labels**
  - **Train another binary classifier which takes y as input and predict whether x is in the f's training dataset or not (membership inference attacks)**

J3– Membership Inference Attacks

- **Project Goal:**
  - ➢ Relate the success of membership inference attacks with the overfitting level of the model (e.g., the generalization gap)

- **Project Outline:**
  - ➢ Study the relationship between generalization gap, model accuracy (testing accuracy of f), and the success rate of membership inference attacks (is there a tradeoff between utility and privacy?)
  - ➢ Design new membership inference attacks.
  - ➢ Evaluate them on benchmark datasets.
  - ➢ Discuss the possible defenses.

- **Reference:**
  - Shokri et al. “Membership inference attacks against machine learning models”. In IEEE S&P 2017.

SW1- Model compression with knowledge distillation/quantization (cap: 2)

- **Introduction:**
  - ➢Model compression with knowledge distillation is to training a compact model by exploiting knowledge from the larger model;
  - ➢It may be challenge to retrain the performance as the original model, especially when applying distillation techniques to multi-class object detection, in contrast to image classification.
  - ➢Model compression with quantization is to apply quantization techniques to compress the storage of deep neural networks;
  - ➢Quantization techniques may include binarizing the parameters, scalar quantization using K-means, structured quantization using product quantization.

SW1- Model compression with knowledge distillation/quantization

- **Project Goal:**
  - ➢The goal is to maintain a similar performance while reducing a number of redundant parameters/flops of the network.
- **Project Outline:**
  - ➢Review model compression algorithms with knowledge distillation/quantization.
  - ➢Propose your method/algorithms.
  - ➢Evaluate your method on various deep neural architectures, and you may want to focus on the image classification task first.
  - ➢Compare with other methods or compact models.
- **Reference:**

[1] A. Romero, N. Ballas, S. Kahou, A. Chassang, C. Gatta and Yoshua Bengio, "FitNets: Hints for Thin Deep Nets," 2015. https://arxiv.org/abs/1412.6550.

[2] Y. Gong, L. Liu, M. Yang and L. Bourdev, "Compressing Deep Convolutional Networks using Vector Quantization," 2014. https://arxiv.org/abs/1412.6115.

[1] A. Polino, R. Pascanu and D. Alistarh, "Model compression via distillation and quantization," 2018. https://arxiv.org/abs/1802.05668.

[2] Y. Cheng, D. Wang, P. Zhou and Tao Zhang, "A Survey of Model Compression and Acceleration for Deep Neural Networks," 2017. https://arxiv.org/abs/1710.09282.

[3] Model-Compression-Papers, https://github.com/chester256/Model-Compression-Papers.

MS1- Neural Architecture Search (NAS) in GAN (cap: 2)

- **Introduction:**
  - ➢NAS is referred to as automating the design of deep neural networks.
  - ➢Recently, there are many approaches based on Bayesian optimization, Reinforcement learning, gradient based methods for NAS.
  - ➢GAN is a recent deep generative model with remarkable performance in generating realistic images.
  - ➢In general, GAN needs two architectures for the discriminator and generator networks.
  - ➢Current approaches for designing both networks are based on trial and error.

MS1- Neural Architecture Search (NAS) in GAN

- **Project Goal:**
  - The goal of this project is to use NAS techniques to automate designing the generator and discriminator architectures.
- **Project Outline:**
  - Reviewing NAS techniques
  - Reviewing the GAN concept briefly
  - Presenting your proposed method
  - Evaluating your method on some real and relatively simple data set (preferably CIFAR-10)
  - Compare with at least one existing method
- **Reference:**
  - **https://openaccess.thecvf.com/content_ICCV_2019/html/Gong_AutoGAN_Neural_Architecture_Search_for_Generative_Adversarial_Networks_ICCV_2019_paper.html**

MS2- Unstructured Pruning of Deep Neural Models (cap: 2)

- **Introduction:**
  - Deep neural networks are computationally intense.
  - They need a large memory for saving trained weights.
  - Memory requirement and computation load make them difficult to be deployed in hard-ware limited devices.
  - Model compression techniques try to comperes a deep neural model without sacrificing the performance of the full model.
  - Among compression methods, pruning techniques are the ones achieve compressed model with removing less important weights/kernels/layers.
  - Unstructured pruning methods prune redundant weights.

MS2- Unstructured Pruning of Deep Neural Models

- **Project Goal:**
  - ➢ The goal of this project is to explore less important weights based on different criteria and removing them with some efficient algorithm.
- **Project Outline:**
  - ➢ Reviewing unstructured pruning techniques and explain how different these methods are compared to the structured methods
  - ➢ Reviewing what measures are used to determine the less important weights
  - ➢ Presenting your proposed method
  - ➢ Evaluating your method on some real and relatively simple data set and task (preferably CIFAR-10 and classification)
  - ➢ Compare with at least two existing methods
- **Reference:**
  - ➢ https://arxiv.org/pdf/1710.09282.pdf
  - ➢ https://github.com/chester256/Model-Compression-Papers

MS3- Audio Separation Using Deep Neural Networks (cap: 2)

- **Introduction:**
  - Audio separation is a classical problem in speech processing.
  - Here the goal is to demix (separate) each sources of speech from their mixed signal.
  - We have seen this application in ICA.
  - However, ICA is a linear model which might not be satisfactory in many scenarios.

MS3- Audio Separation Using Deep Neural Networks

- **Project Goal:**
  - The goal of this project is to use nonlinear technique such as deep learning to separate audio sources from their mixed signal.
  - The mixed signal may be mixture of music background, dog barking, and siren voice.
- **Project Outline:**
  - Reviewing deep learning methods for source separation (demixing)
  - Preparing training and test data sets. Specifically, you need to provide a detail of preprocessing steps including chunking, sampling, and formatting of data
  - Presenting your proposed method
  - Evaluating your method on your prepared data set.
  - Compare with at least the ICA and the method proposed by:
    - **http://cs230.stanford.edu/projects_fall_2019/reports/26261998.pdf**
- **Reference:**
  - **http://cs230.stanford.edu/projects_fall_2019/reports/26261998.pdf**

MS4 – Designing Recommender Systems for Restaurant Data with Consumer Ratings (cap: 2)

- **Introduction:**
  - Recommender systems are very useful tool for many businesses.
  - For example, Amazon uses your history of orders to suggest you new items.
  - For example, Netflix uses your history of watched movies to suggest your favorite movies.
  - There are many classical approaches for designing recommender systems, including collaborative filtering.
  - Fundamentally, matrix completion problem is an example of recommender systems.

MS4 – Designing Recommender Systems for Restaurant Data with Consumer Ratings

- **Project Goal:**
  - ➢ The goal of this project is to design a deep learning approach to output top list of restaurants according to the consumer preferences
- **Project Outline:**
  - ➢ Reviewing non-deep learning methods for designing recommender systems
  - ➢ Reviewing deep learning methods for designing recommender systems
  - ➢ Presenting your proposed method. Your method can be generative or discriminative
  - ➢ Evaluating your method on the data set provided by Kaggle:
    - **https://www.kaggle.com/uciml/restaurant-data-with-consumer-ratings**
- **Reference:**
  - ➢ **https://www.kaggle.com/uciml/restaurant-data-with-consumer-ratings**

MS5 – Deep learning Approach for Question and Answering (QA) Systems (cap: 2)

- **Introduction:**
  - ➢Question answering (QA) is a classical problem in NLP.
  - ➢The goal of QA systems is to simulate human conversation by developing dialog systems and chatbots.
  - ➢Traditional methods for designing QA systems are based on parsing, part-of-speech tagging and coreference resolution.
  - ➢Recent progress in Recurrent Neural Networks (RNNs) make them a popular candidate for designing QA systems.

MS5 – Deep learning Approach for Question and Answering (QA) Systems

- **Project Goal:**
  - ➢ The goal of this project is to design a QA system using deep learning for **bAbI** data set .
- **Project Outline:**
  - ➢ Reviewing non-deep learning methods for designing QA systems
  - ➢ Reviewing deep learning methods for designing QA systems
  - ➢ Presenting your proposed method based on *attention* mechanism
  - ➢ Evaluating your method with a base-line seq-to-seq method
- **Reference:**
  - ➢ **https://research.fb.com/downloads/babi/** **(bAbI)**

MA1 – Action Recognition from Videos (cap: 2)

- **Introduction:**
  - A video can be viewed as a time sequence of 2D images that exhibit temporal correlation
  - An interesting problem of significant practical importance is the recognition of various different human activities from videos
    - Various different datasets are publicly available
    - Preferred example: THETIS data set comprises videos of 12 basic shots in tennis performed by professionals and amateurs; the objective would be to recognize the shot in a given video
    - Other examples: UCF101 (13320 videos, 101 actions), HMDB51 (7000 clips, 51 actions) datasets
  - Conventional methods for action recognition from videos have been developed as a straightforward generalization of single-image CNN methods, and use 3D CNN modules
    - Have proven to be successful, but do not scale well
  - Recent developments suggest combining multiple models
    - Algorithms search for best hyper-parameters of a combined architecture of models

MA1 – Action Recognition from Videos

- **Project Goal:**
  - ➢ Develop and implement a DL model for tennis shot recognition (THETIS dataset)
    - ➢ As an alternative, other datasets (e.g., UCF101, HMDB51) can be also used
- **Project Outline:**
  - ➢ Overview of state-of-art methods for deep learning in videos
  - ➢ Description of THETIS database (or database of your choice) and overview of current benchmarks
  - ➢ Propose and evaluate your own design of DL model
- **Reference:**
  - http://thetis.image.ece.ntua.gr/
  - http://openaccess.thecvf.com/content_cvpr_2017/papers/Feichtenhofer_Spatiotemporal_Multiplier_Networks_CVPR_2017_paper.pdf
  - https://arxiv.org/pdf/1811.10636.pdf

MA2 – Robust Neural Decoding from Limited EEG data (cap: 2)

- **Introduction:**
  - Brain-computer interfacing is one of the most exciting technologies of the future
    - The goal in BCIs is to predict the motor action a subject intends to perform from neural recording (such as EEG for instance)
    - Applications in healthcare, civilian and public domain, as well as the tactical domain
  - One common issues in BCIs is the limited data
    - The limited data constraints the applicability of deep learning methods in BCIs
    - Prior work has heavily relied on simplistic approaches ML approaches for neural decoding that involve heuristic feature extractors from brain signals and simple classifiers
  - Recent insights suggest that robust estimation methods can help extract relevant features, allowing deep model to be trained from limited data
    - However, it is still unclear whether these methods are of any help in EEG-based BCIs

MA2 – Robust Neural Decoding from Limited EEG data

- **Project Goal:**
  - Investigate whether robust feature extraction methods can drive applicability of neural networks methods and improve performance of neural decoders based on EEG

- **Project Outline:**
  - Literature overview and description of available databases
    - Familiarization with robust estimation
  - Implementation of a neural decoding pipeline with robust feature extractors and DL-based classifier and evaluation over EEG datasets

- **Reference:**
  - https://bmcbiomedeng.biomedcentral.com/articles/10.1186/s42490-019-0022-z
  - https://arxiv.org/abs/1901.10397

MA3 – Deep Learning for Wildfire Detection (cap: 3)

- **Introduction:**
  - Wildfires are becoming a major problem as we speak
    - The response delay minimization is crucial for timely wildfire mitigation
  - Wildfires patterns are driven by multiple factors
    - Location, weather (climate), season, human activity ect.
  - We can leverage deep learning to detect wildfires early on
  - Comprehensive wildfire data in the US can be obtained through the Wildland Fire Open Data project from the National Interagency Fire Center
    - Formerly known as GeoMAC
  - The database is **multimodal**
    - contains location, perimeter polygons and date of all wildfires in the United States (in-going and past)

MA3 – Deep Learning for Wildfire Detection

- **Project Goal:**
  - ➢ Explore deep learning approaches for detecting wildfires

- **Project Outline:**
  - ➢ Literature overview
  - ➢ Data formatting and manipulation
  - ➢ Propose and evaluate deep learning method for detecting wildfires
  - ➢ Extra challenge: combine the wildfire data with public satellite imagery data

- **Reference:**
  - https://data-nifc.opendata.arcgis.com/
  - https://www.youtube.com/watch?v=Ch2HQo8mhGo&feature=youtu.be&ab_channel=NIFCFireAviation

MA4 – Object Detection on xView (cap: 3)

- **Introduction:**
  - xView is one of the largest publicly available datasets of overhead imagery
  - Contains satellite images from complex scenes
    - 1 million objects, 60 classes, 0.3 meters of resolution, more than 1400 km^2
  - There are some inherent challenges in xView
    - Large class-imbalance
    - Small objects
    - Densely packed objects

MA4 – Object Detection on xView

- **Project Goal:**
  - ➢ Develop software for object detection on the xView dataset using DL methods

- **Project Outline:**
  - ➢ Overview of benchmark architectures and results
  - ➢ Implementation of **your own** model for object detection on the xView

- **Reference:**
  - http://xviewdataset.org/
  - https://arxiv.org/abs/1802.07856
  - https://arxiv.org/abs/1903.01347

Comprehensive Reviews

Presentation format: Beamer class

1. Bayesian Neural Networks (2 students, 80-100 slides)
   - Gaussian Process and Deep Neural Models
   - Neural Tangent Kernel
   - Practical Bayesian Deep Learning

2. Deep Learning Techniques and Inverse Problems (1 student, 40-50 slides)
   - Tikhonov regularization and deep learning
   - Applications (to imaging)

Comprehensive Reviews

3. Meta-Learning (2 students, 80-100 slides)
   - Few-Shot Learning
   - Metric Learning
   - Recurrent Model Learning

4. Adversarial Robustness of Deep Learning (2 students, 80-100 slides)

5. Interpretability/Explainability in Deep Learning (2 students, 80-100 slides)