**PROJECT PLAN**

**By submitting this plan, you confirm that you have access to the required computational resources and tools to execute this plan. This plan (or any subsequent change) will be reviewed, to ensure that all plans have a similar level of complexity. If your plan is not reviewed, it might likely not be at the level of what we expect, which will end up impacting your grade. Thus, it is in your best interest to have your plan reviewed.**

1. Student names. (The project is to be done in groups of 3 students.)

* Pranav Pai, Shriya Kumar, Gaurav Asok Kumar (Group 10)

1. [Up to 5 lines] Definition of the problem and research question, possibly relevant to your interests.

* **Problem:** Detecting prohibited items in X-ray images is challenging, especially with occlusion. While STNs (Spatial Transformer Networks shown promise in improving detection through adaptive transformations, their performance across varying occlusion levels needs further investigation.
* **Research Question:** Can STNs, when combined with different neural networks, be adapted to improve the detection and localization of prohibited items in X-ray images with varying occlusion levels?

1. [Up to 5 lines] Description of the dataset (or datasets) to be used, e.g., number of tables, rows, columns, type of data (discrete/categorical, continuous, sentences, images, etc.). Datasets should be already publicly available since there is not enough time for you to collect data. Possible datasets are for instance: [ADHD 200 (Whole Brain Data)](http://www.nitrc.org/plugins/mwiki/index.php/neurobureau:AthenaPipeline#Whole_Brain_Data), [Labeled Faces in the Wild](http://vis-www.cs.umass.edu/lfw), [Heritage Health Prize](https://www.kaggle.com/c/hhp), [Yahoo Bidding (A1)](http://webscope.sandbox.yahoo.com/catalog.php?datatype=a), [Yahoo Ranking (C14)](http://webscope.sandbox.yahoo.com/catalog.php?datatype=c). **You should choose a challenging dataset. Small datasets are not allowed. (Please also see question 5).**

* We use a subset (1%) of the SIXray dataset, which contains **192\* images of X-ray scans (training)** and 58\* images of the test set (7%) (.jpg). Each image has a corresponding **XML annotation file** that provides the object labels (e.g., guns, knives), bounding box coordinates (xmin, ymin, xmax, ymax), and image dimensions (width, height, depth). The data is continuous (images), with annotations for categorical object classes. The dataset is challenging, particularly due to the presence of multiple objects with various levels of occlusion, making it suitable for complex object detection tasks.
* *\*Caveat: We started out using a bigger dataset (~7,514 training images and ~836 test images) and ran all steps up until modelling, however, we faced severe computational restrictions and GPU limitations since training large amounts of images is very computationally expensive*

1. URL where the above dataset(s) is(are) available.

* **https://github.com/MeioJane/SIXray?tab=readme-ov-file**

1. [up to 5 lines] Which feature construction and preprocessing of the dataset will be performed, e.g., converting several tables to a single table, counting, summing, one-hot encoding, etc. **You are not allowed to just read a single data table and use it. Remember this is a Masters level subject and we require some level of complexity. You are allowed to either implement this from scratch or use third-party code.**

* We will perform several preprocessing steps:
  + Resizing
  + White Space Removal for all images is done dynamically. This step is implemented on our own without any external libraries.

We will perform several data augmentation steps (added synthetic occluded images):

* + Flipping (hflip, vflip, hvflip)
  + Colour Jitting (random brightness, contrast, saturation and hue of an image)
  + Gaussian Blurring (For the images)
  + Gaussian Noise (For the whole image / also implemented only within the Bounding boxes)
  + Random EraAsing for images (Random Erasing randomly selects a rectangle region in an image and erases its pixels with random values - also implemented within the bounding boxes)
  + Random blocking and blending (blocking portions of the image randomly and then blending for seamless integration of the blocks)

We will perform several feature-generation steps:

* One-Hot encoding
* Normalising
* Occlusion levels generation

1. [Up to 5 lines] Which 3 machine learning algorithms are going to be compared? You should list 3 different algorithms and with different model class complexity, i.e., simple, medium, complex. **You are allowed to either implement this from scratch or use third-party code. At least one of the 3 algorithms should be from a research paper in a conference or journal, e.g.,** [**NeurIPS**](https://proceedings.neurips.cc/)**,** [**ICML/UAI/AISTATS/JMLR**](https://proceedings.mlr.press/)**,** [**ICLR**](https://openreview.net/group?id=ICLR.cc)**,** [**TMLR**](https://openreview.net/group?id=TMLR)**, etc. (Since this research-paper algorithm might take you more time to figure out, you can let us know this later.)**

* **Faster -RCNN (**Faster Regional-Convolutional neural network**)**
  + <https://ieeexplore.ieee.org/abstract/document/8243900> **- (Faster RCNN)**
* **Faster -RCNN + STN (**Faster Regional-Convolutional neural network **+** Spatial Transformer Networks**)**
  + <https://arxiv.org/abs/1506.02025> (Also published in NeurIPS) - (**Spatial Transformer Networks)**
* **RFB Net** (Receptive Field Block net)
  + [**https://arxiv.org/abs/1711.07767**](https://arxiv.org/abs/1711.07767) **- (Receptive Field Block Net for Accurate and Fast Object Detection)**

**Why STNs?**

STNs provide a unique advantage by allowing networks to learn spatial transformations (like rotation, scaling, and translation) internally, making them ideal for handling the complex spatial variability seen in X-ray images. Unlike traditional networks that rely purely on feature extraction, STNs enable our models to correct distortions and occlusions actively, improving detection accuracy even when prohibited items are partially hidden or misaligned.

**Faster R-CNN (Baseline)**:  
Faster R-CNN combines region proposal generation and object detection in a unified framework, allowing for accurate detection and localisation of objects. It's effective for detecting prohibited items in X-ray images, making it a strong baseline for comparison.

**Faster -RCNN +STN (**Faster Regional-Convolutional neural network +Spatial Transformer Networks**)**:  
While traditionally, CNN combined with STN helps the model adaptively focus on key areas in the image by applying spatial transformations, we apply this same logic to Faster RCNN since it deals with multi-class object detection with occlusions

**RFB Net:**:  
RFB Net enhances multi-scale object detection by incorporating different receptive field sizes. We expect the model to perform in complex scenarios with varied occlusion levels.

1. [Up to 5 lines] Cross-validation technique, e.g., training/validation/testing, k-fold cross-validation, bootstrapping. **You MUST implement this from scratch. At least 20 independent repetitions should be run, so that means/variances can be computed for proper comparison between algorithms.**

* 20-fold cross validation.

1. [Up to 10 lines] Which hyperparameter(s) is(are) going to be tuned for each of the 3 algorithms above, and what method is going to be used for the nested cross-validation? **You MUST implement this from scratch. Every algorithm should have at least one hyperparameter to be tuned and such hyperparameter should be expected to affect the results significatively.**

* **Faster RCNN:** Anchor box sizes/ratios and RPN non-maximum suppression (NMS) threshold.
* **Faster RCNN + STN:** We select one STN type (affine) before tuning and tuning theanchor box sizes keeping other Faster RCNN elements consistent with what we find for algo 1.
* **RFB Net:** The number of RFB layers and the aspect ratio of the receptive fields will be tuned.

For each algorithm, we will use **Stratified nested k-fold cross-validation**, to ensure balance between images with different classes. Where the inner loop performs hyperparameter tuning (grid search or random search) and the outer loop evaluates model performance on held-out data.

1. [Up to 15 lines] Description of the experimental results, e.g., learning curves, ROC curves, plots of different datasets, etc. **You MUST implement this from scratch. Error bars should be computed across repetitions (at least 20 as described in question 7) and reported for proper comparison between algorithms.**

**Experimental Results.**

* **Mean Average Precision (mAP)**: Measures how well the model detects and classifies objects at different confidence levels. mAP considers both precision and recall for each class.
* **Precision and Recall:** This can be integrated with mAP to calculate P&R.
* **F1-Score**
* **Intersection over Union (IoU):** This metric evaluates how well the predicted bounding boxes overlap with the ground truth. Higher IoU values indicate better localization of detected objects.
* **Training Time and Inference Speed:** Measure how fast our model trains and predicts. Since efficiency is a key factor for real-time applications like X-ray screening,
* **Error Bars are** computed across repetitions during k-fold cross-validation.

**What would we be testing and how is it related to our question?  
  
First test on our main test dataset (Original unseen test data from the source)**

For each image in the dataset, we create a measure to quantify the level of occlusion (overlapping) in the image. Using this occlusion index we identify how the model is performing with images of different occlusion levels.

1. Which programming language are you going to use? (Python, Jupyter, C++, MATLAB, and Java are allowed.)

* Python