ML Study Design - Google Street View Blurring System

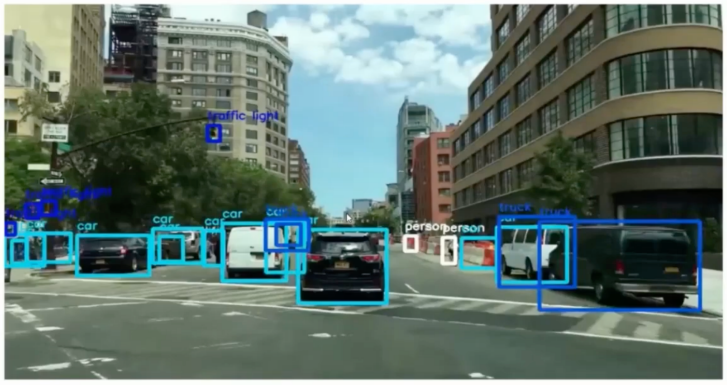
Objective:

ML System Design -02
Google Street View Blurring System

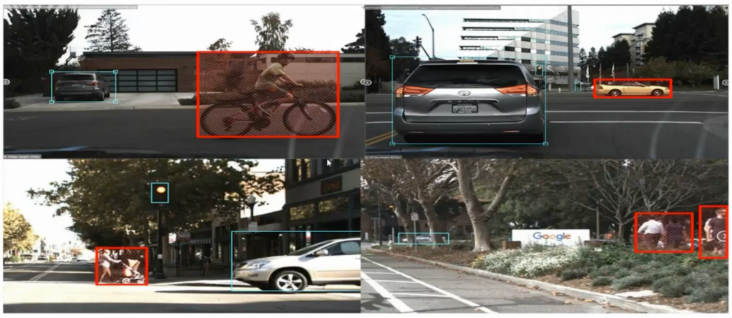
For Privacy protection blurring faces & Car License plates

1. Clarifying the Requirements: Designing a Street View Blurring System which blurs human faces and license plates. B.O. is to protect user privacy. We will also be using training dataset of 1M annotated images of human faces and license plates.

2. Framing the problem as an ML task: To accurately detect object of interest in the image. After detecting the objects we can blur them before displaying it to users.



A. Specifying the i/o of the system: An image with zero to multiple objects at different locations within it. The model would detect and outputs those locations.



In a given image, first objects are located and then the bounding boxes are created around the object.

3. Choosing the right ML category.

1. Predicting the location of Object in the image.
2. Predicting the class of each bounding box (person, car, people on bike/cycle etc.)

- The first is regression problem, where the location can be specified by (x,y) coordinates. The second one is multi-class classification.
- Generally, object detection are divided in two parts:
  1. Two Stage Network
  2. One Stage Network

1. Two stage Network: (RCNN, Fast RCNN, Faster RCNN)
   1. Region Proposal Network (RPN).
   2. Classifier.

2. One-Stage Network: (YOLO, SSD)
   - Both the stages are combined. Bounding boxes and object classes are generated simultaneously, without explicit detecting for regional proposals.

What is an RPN?

A **Region Proposal Network (RPN)** is a critical component in object detection systems like Faster R-CNN. Its role is to generate **proposals**—regions in an image that are likely to contain objects—quickly and accurately.

How Does It Work?

1. **Feature Extraction**:
    - The RPN uses a **feature map** produced by a convolutional neural network (CNN) as input.
    - This feature map captures essential details about the image, such as edges, textures, and patterns.
2. **Anchor Boxes**:
    - At each position in the feature map, RPN places predefined **anchor boxes**—rectangles of various sizes and aspect ratios.
    - These boxes act as potential candidates for object regions.
    - **Example**: For a 10x10 feature map with 3 scales and 3 aspect ratios, RPN creates 10×10×9=900anchor boxes.
3. **Objectness and Bounding Box Refinement**:
    - For each anchor box, RPN predicts:
        - **Objectness Score**: How likely it is that the box contains an object (vs. background).
        - **Bounding Box Adjustment**: Precise shifts to the box's position, width, and height to better match the object.
    - **Example**: An anchor box at (5, 5) with size 32x32 pixels might be refined to:
        - Objectness score: 0.9 (high confidence)
        - Adjusted box: Center (5.2, 5.1), Size 30x40 pixels.
4. **Proposal Generation**:
    - **Filtering**: Anchor boxes with low objectness scores are discarded.
    - **Refinement**: The remaining boxes are adjusted using the bounding box predictions. An anchor box might be at (x=100, y=150) with width 50 and height 30. The RPN might predict dx=2, dy=-1, dw=4, dh=-2. The refined box would then be at (x=102, y=149) with width 54 and height 28.
    - **Non-Maximum Suppression (NMS)**: Overlapping proposals are merged, leaving only the highest-quality ones.

        How it works:

        - Sort the remaining proposals by their objectness scores in descending order.
        - Select the proposal with the highest score.
        - Compare this proposal with all other remaining proposals.
        - If the Intersection over Union (IoU) between the selected proposal and any other proposal is greater than a certain threshold (e.g., 0.7), discard the other proposal (because it's considered redundant).
        - Repeat steps 2-4 until all proposals have been considered.
        - **Intersection over Union (IoU):** IoU is the ratio of the area of overlap between two bounding boxes to the area of their union. A high IoU means the boxes overlap significantly.
        - **Effect:** NMS ensures that each object is represented by only one (or a few) high-quality proposal(s), further reducing the number of proposals passed to the next stage of the object detection pipeline.

Why is RPN Important?

- RPN enables **end-to-end training**, allowing the model to jointly optimize region proposals and object classification.
- It's computationally efficient, as it operates directly on the feature map without requiring separate sliding window or region extraction steps.

YOLO

YOLO (You Only Look Once) is a single-stage object detection system that performs detection directly on a grid overlaid on the input image. Here's a concise breakdown:

1. **Grid Division:** The input image is divided into an S x S grid (e.g., 7x7).
2. **Cell Predictions:** Each grid cell is responsible for predicting:
    - **B bounding boxes:** Each bounding box has:
        - (x, y): Center coordinates relative to the cell.
        - (w, h): Width and height relative to the image.
        - Confidence score: Probability of an object being in the box *and* the box being accurate.
    - **C class probabilities:** A probability distribution over the C object classes.
3. **Encoding:** These predictions are encoded into a tensor of size S x S x [B * 5 + C], where 5 represents the 4 bounding box coordinates + the confidence score.
4. **Non-Max Suppression (NMS):** Because multiple cells might detect the same object, NMS is used to filter out redundant bounding boxes, keeping only the ones with the highest confidence scores and sufficient IoU (Intersection over Union).
5. **Loss Function:** YOLO uses a loss function that combines:
    - Bounding box regression loss (how well the predicted boxes match the ground truth).
    - Confidence loss (how accurate the objectness predictions are).
    - Classification loss (how accurate the class predictions are).

Key Differences from Two-Stage Detectors:

- **No Region Proposals:** YOLO directly predicts bounding boxes and class probabilities without a separate region proposal step. This is what makes it much faster.
- **Grid-Based Detection:** Detection happens at the grid cell level. Each cell is responsible for predicting objects whose centers fall within it.

One Stage vs Two Stage:

- Two stage networks perform the operation in two sequential steps (slower but accurate).

We will work with Two Stage.
We are dealing with dataset of 1 million images which is not huge by modern standards. And when the training data increases or the need for more quicker predictions arises in future, we can later shift to One Stage.

Some other models

Model	Type	COCO mAP (%)	Description
YOLOv10	Single-Stage	~55.0	Latest iteration of the YOLO family, optimized for speed and accuracy, outperforms earlier versions.
YOLOv9	Single-Stage	~54.0	Improved version of YOLO, focused on better performance and efficiency.
YOLOv8	Single-Stage	~52.0	Enhances the YOLO framework with better optimization for real-time applications.
YOLOv7	Single-Stage	~50.0	Prior iteration in the YOLO series, offering a balance of speed and accuracy.
Cascade R-CNN	Two-Stage	~52.9	Extends Faster R-CNN with multiple stages for refined localization and classification.
DEYO	Two-Stage	52.1	Combines DETR's transformer-based detection with YOLO's efficiency for enhanced performance.
AFDetV2	Single-Stage	N/A	Anchor-free detector achieving state-of-the-art results on Waymo Open Dataset and nuScenes Dataset.
RTMDet	Single-Stage	~50.0	Real-time detector balancing speed and accuracy for practical applications.
ViTDet	Two-Stage	~49.5	Leverages Vision Transformers for object detection with competitive accuracy.
DETR	Two-Stage	~44.9	Introduces transformers into object detection, providing a novel approach to feature extraction.

Data Preparation:Data Engineering:

- Annotated Dataset.
- Street View Images.

1. Annotated Datasets: We have 1 million annotated images with each having bounding boxes and associated Object classes.

Image path	Objects	Bounding boxes
dataset/image1.jpg	human face human face license plate	[10, 10, 25, 50] [120, 180, 40, 70] [80, 95, 35, 10]
dataset/image2.jpg	human face	[170, 190, 30, 80]
dataset/image3.jpg	license plate human face	[25, 30, 210, 220] [30, 40, 30, 60]

2. StreetView Images:

The ML system will process these images to detect human faces and license plates.

Image path	Location (lat, lng)	Pitch, Yaw, Roll	Timestamp
tmp/image1.jpg	(37.432567, -122.143993)	(0, 10, 20)	1646276421
tmp/image2.jpg	(37.387843, -122.091086)	(0, 10, -10)	1646276539
tmp/image3.jpg	(37.542081, -121.997640)	(10, -20, 45)	1646276752

Data Engineering : Feature Engineering

- Applying standard Image Preprocessing operations like Resizing, normalization etc. we will be using data augmentation to increase the size of the dataset.

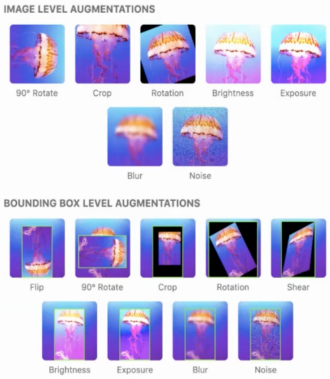
Data Augmentation: Adding a slightly modified copies of original images or creating new images from the original image.

- Helps learn complex patterns
- Helps with imbalanced dataset.

Careful with rotation, flipping and bounding boxes In rotation.

Performing Augmentation: Offline vs Online

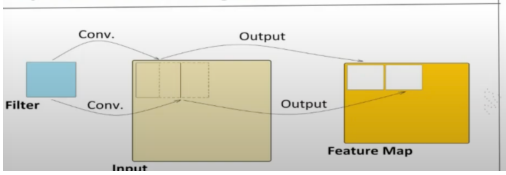
Offline - Fast training time but takes more storage
Online - Slow training time but doesn't take additional storage.

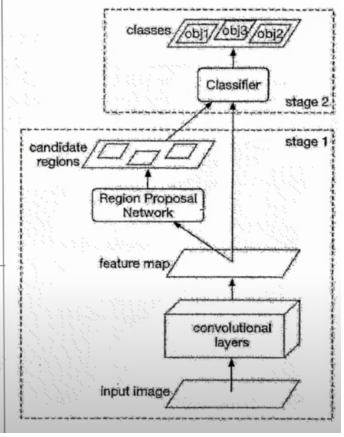


Model Development:
Convolutional Layer - prepares feature map of image by taking in input as an image.

RPN - proposes candidate regions, takes feature map as input and gives candidate regions in the image.

Classifier - Takes the feature map and the proposed candidate regions and assigns an object class to each region.





Model Training:

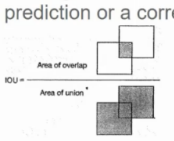
- The model is expected to do two tasks well:
  - The bounding boxes are supposed to highly overlap the ground truth bounding boxes.
  - Predicted probabilities for each object should be accurate.
- We will use regression loss and classification loss - loss functions.

Usually contains: Forward propagation, Backward propagation and loss calculation.

Evaluation:
Generally, the machine learning model has to detect N different objects in the image. So to evaluate the model's performance, we will evaluate each object separately and average the results.

Intersection Over Union (IOU):

- Measures the overlap of the bounding boxes.
- Shows the detected bounding box are aligned with ground truth bounding box.
- IOU=1 indicates they are fully aligned, though rare.
- Higher IOU means more accuracy.
- IOU higher than 0.7 is considered a good prediction or a correct detection.



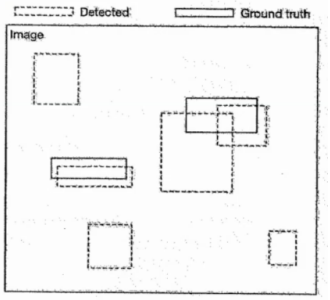


Figure 3.8: Ground truth and detected bounding boxes

Example of per-class evaluation and mAP for a two-stage detector:

Two classes ("cat", "dog") and two images are used for a simplified example.

- **Image 1:** 2 cats, 1 dog (ground truth). Predictions: 3 cat predictions (2 correct), 1 correct dog prediction.
- **Image 2:** 1 cat, 2 dogs (ground truth). Predictions: 1 correct cat prediction, 3 dog predictions (2 correct).

Using precision and recall (simplified, AP is usually from a curve):

- **Cat:** Image 1: P=0.67, R=1.0. Image 2: P=1.0, R=1.0. AP (simplified average): 0.835
- **Dog:** Image 1: P=1.0, R=1.0. Image 2: P=0.67, R=1.0. AP (simplified average): 0.835

mAP (average of per-class APs): (0.835 + 0.835) / 2 = 0.835

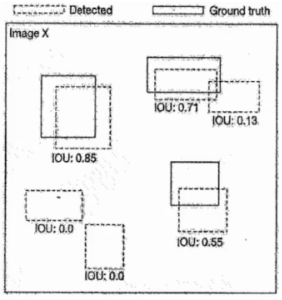
Offline metrics:

Precision: P = correct precision/ number of precision
But for different IOU threshold eg. =0.7, 0.5 and 0.1 the value changes.

Average Precision: for measuring the precision of single object being detected by the model.

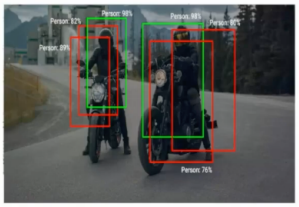
mAP : for measuring the model's overall performance.

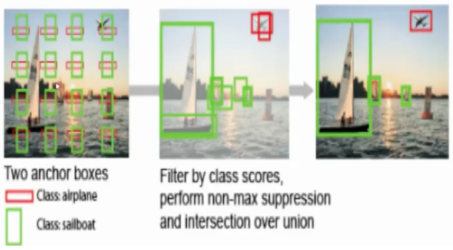
Online metrics:
We will use "User Reports".

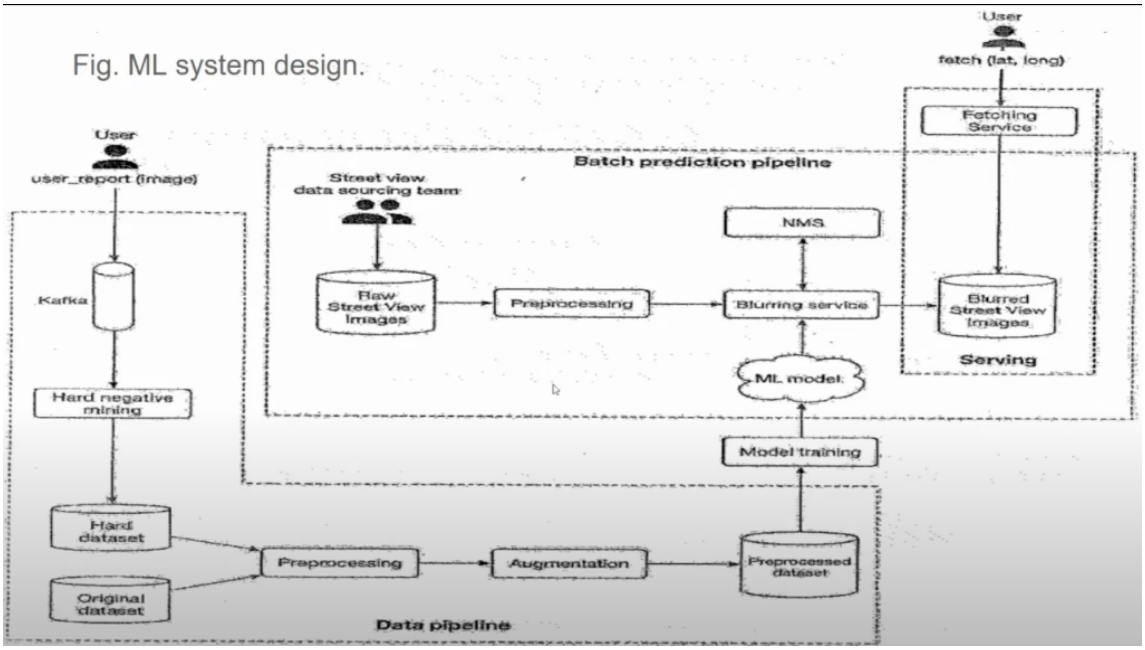


Serving:

- When we run an object detection algorithm, it is common to see overlapping of bounding boxes because RPN proposes various regions in the image.
- So, it is important to bring that down to one bounding box. For that we will use an algorithm, NMS.
- NMS will keep highly confident boxes and will remove the overlapping boxes.







The "Hard negative mining" component likely processes the initial training results to identify and select the most challenging false positives. These hard negatives are then used to augment the dataset for subsequent training iterations.

Batch Prediction pipeline:

1. Raw Street View images.
2. Preprocessing - preparing the images for the model.
3. Blurring system:
    a. Provides a list of detected objects.
    b. NMS gives the final detections.
    c. Blurs detected object.
    d. Stores the image in object storage.
    - The first two processes a & b are CPU bound processes, the next two processes c & d are GPU bound processes.
    - Advantages :
        - Scaling the services independently.
        - Better utilization of CPU and GPU.

Data Pipeline:

User-reports - we collect reports from user, and from hard negative mining we produce a hard dataset.

Combining those with the original dataset we train the model to improve the performance.

Questions:

Dr. Mamta's Question:
●**Question:** Is there a way to understand the data and what kind of data augmentation can be applied without misunderstanding the data, especially when there is limited data and human intervention is undesirable?
●**Answer:** Jit acknowledged that data augmentation needs careful consideration. He explained that in the Street View example, using completely upside-down images would be inappropriate because they wouldn't occur in reality. He suggested focusing on realistic scenarios and data that would likely be encountered. Prasa and Dr. Mamta highlighted the importance of human intervention, particularly with limited datasets, to ensure appropriate augmentation and avoid inaccurate results. They emphasized the need for a "human in the middle" to evaluate and select suitable data for training to prevent false positives and negatives.

Pry's Questions:
●**Question:** How do you determine what to blur when trying to protect privacy, considering that blurring the whole face might be excessive?
●**Answer:** Jit did not directly address this question. However, Prasa and he agreed that blurring specific features like eyes and noses could suffice for de-identification while reducing processing demands.
●**Question:** How does the system differentiate overlapping objects, like a person in a vehicle, when applying different blurring to each?
●**Answer:** Jit didn't explicitly answer this. However, his presentation explained that the system uses bounding boxes and assigns object classes (person, car, etc.) to each detected region. This

suggests that the model can distinguish and apply different blurring to overlapping objects based on their classifications.

Single-stage object detection models differentiate overlapping objects by:

1. **Bounding Box Regression**: Predicting separate bounding boxes for each object in the overlap region.
2. **Object Classification**: Assigning class probabilities (e.g., person, vehicle) to each predicted bounding box.
3. **Non-Maximum Suppression (NMS)**: Ensuring that overlapping boxes with lower confidence scores are suppressed, retaining the most confident predictions for distinct objects.
4. **Feature Maps**: Utilizing spatial and contextual features from the image to accurately separate and classify objects even in overlapping scenarios.

●**Question:** Since Google Street View is a live view, how does the system process real-time images and select frames for blurring?
●**Answer:** Jit clarified that Google captures panoramic images and stitches them together to create a route view. This process suggests the blurring happens on static images rather than live video streams. Pry and Prasa discussed the complexities of handling live video streams, proposing frame rate reduction and selective image processing as potential solutions.

Prasa's Questions and Suggestions:
●**Question/Suggestion:** Can we discuss various object detection models, including YOLO (You Only Look Once) and SSD (Single Shot Detection), and compare their strengths and weaknesses?
●**Answer/Response:** Jit briefly mentioned YOLO and SSD as examples of one-stage object detection networks5. Prasa emphasized the need to delve deeper into these models, including RCNN variations (Fast RCNN and Faster RCNN), and compare their architectures and performance characteristics.
●**Question/Suggestion:** Can we discuss the differences between commercial models, open-source models, and models described in research papers?
●**Answer/Response:** This question wasn't answered directly. However, Prasa suggested incorporating this comparison into future discussions to provide a broader perspective on the landscape of object detection models.
●**Question/Suggestion:** Should we discuss ImageNet and COCO image databases, which are widely used in the computer vision community?
●**Answer/Response:** This question was not addressed. However, Prasa's suggestion highlights the value of exploring publicly available datasets to understand how models are trained and evaluated.
●**Question/Suggestion:** Can we discuss the mathematical aspects of image processing, focusing on techniques like data pre-processing and augmentation, and the libraries that support them?
●**Answer/Response:** While the presentation touched upon pre-processing techniques like resizing and normalization, it didn't delve into the mathematical details9. Prasa advocated

exploring the mathematical underpinnings of these techniques and discussing libraries that facilitate their implementation. He proposed exploring topics such as wavelet transforms, contour analysis, and relevant libraries to enhance the understanding of image processing techniques.