# Hackwell 2.0

**Presentation Deck:** Survey Corps
**Date:** Jun 4 2021

# Problem Statement

Video search on the basis of image, objects and characteristics.

## Problem 1

Given an image of a person determine the person from the given set of videos.

## Problem 2

Given a text of features determine that person or object from the set of videos.

# Context

Video search on the basis of image, objects and characteristics.

## Situation 1

There have been riots at a certain location and there are multiple footages.

## Situation 2

There is a long video and you want to specifically search for some specific event happening. Checking out the entire video takes a lot of time.

# Core Idea

Indexing the videos on the basis of people and objects present in the images along with the their characteristics.

# Solution: Part 1

## Image Search using Re Identification

- Generated bounding boxes around objects in frames using YOLO.
- Using image hashing resulted in more than 50% reduction in frames
- Used a CNN model (ResNet-50) pretrained on Imagenet to extract high-level features of the dataset.
- ReID measured using distance metrics like
    - Cosine similarity: Easier and automated thresholding
    - Dot Product: Non-Normalised data and manual thresholding

# Solution: Part 2.1

## Using Image Captioning

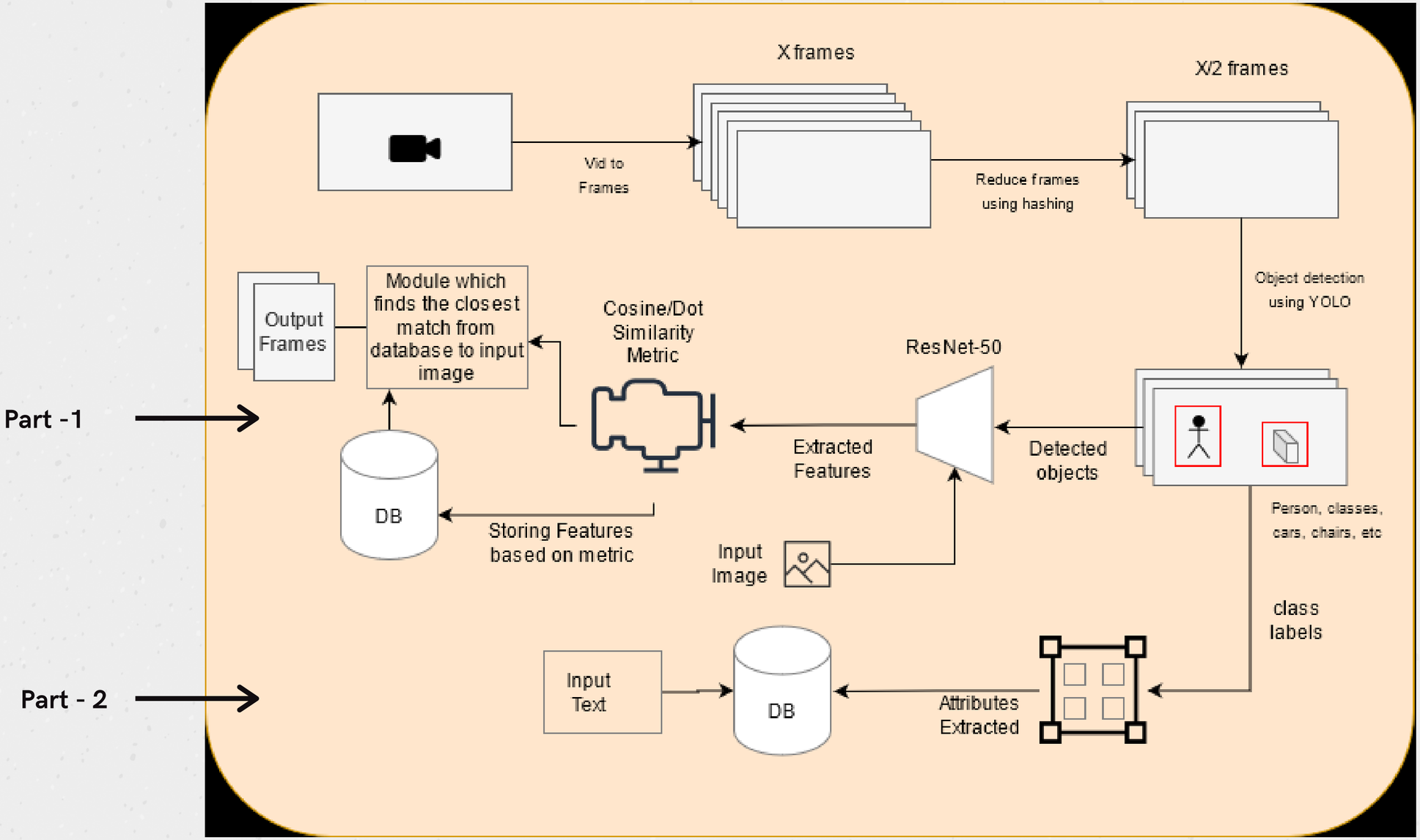
- We tried using image captioning by a CNN-LSTM model trained on Flickr8k dataset
- Since the EPFL dataset doesn't have captions the results weren't the best
- We decided to try something else here

# Solution: Part 2.2

## Image Search using feature vectors

- Used a CNN model (ResNet-50) pretrained on Imagenet.
- Generated bounding boxes around objects in frames using YOLO.
- Reduce the search frames using image hashing.
- Extracted the feature vectors from each detected objects and stored it in a database.
- Input textual features scanned in database, and frames are returned.

# Architecture



# Challenges faced

**Challenge:** A large number of frames present in the video
**Solution:** Used image hashing techniques to optimally find the right set of frames.

**Challenge:** Matching images took a large amount of time
**Solution:** Used the cosine similarity technique with a threshold to classify if the image is similar or not.

# Results

Precision and Speed
Solution- Part: 1

## Precision

Solution 1 had 2 implementations
- Cosine similarity: 76.3%
- Dot Product similarity: 85%

## Speed

From the moment input was passed to the time scanning was done, the entire operation was finished in 0.35 seconds. The search complexity is O(n) where n is the size of database.

# Technologies Used

- PyTorch
- OpenCV
- PIL

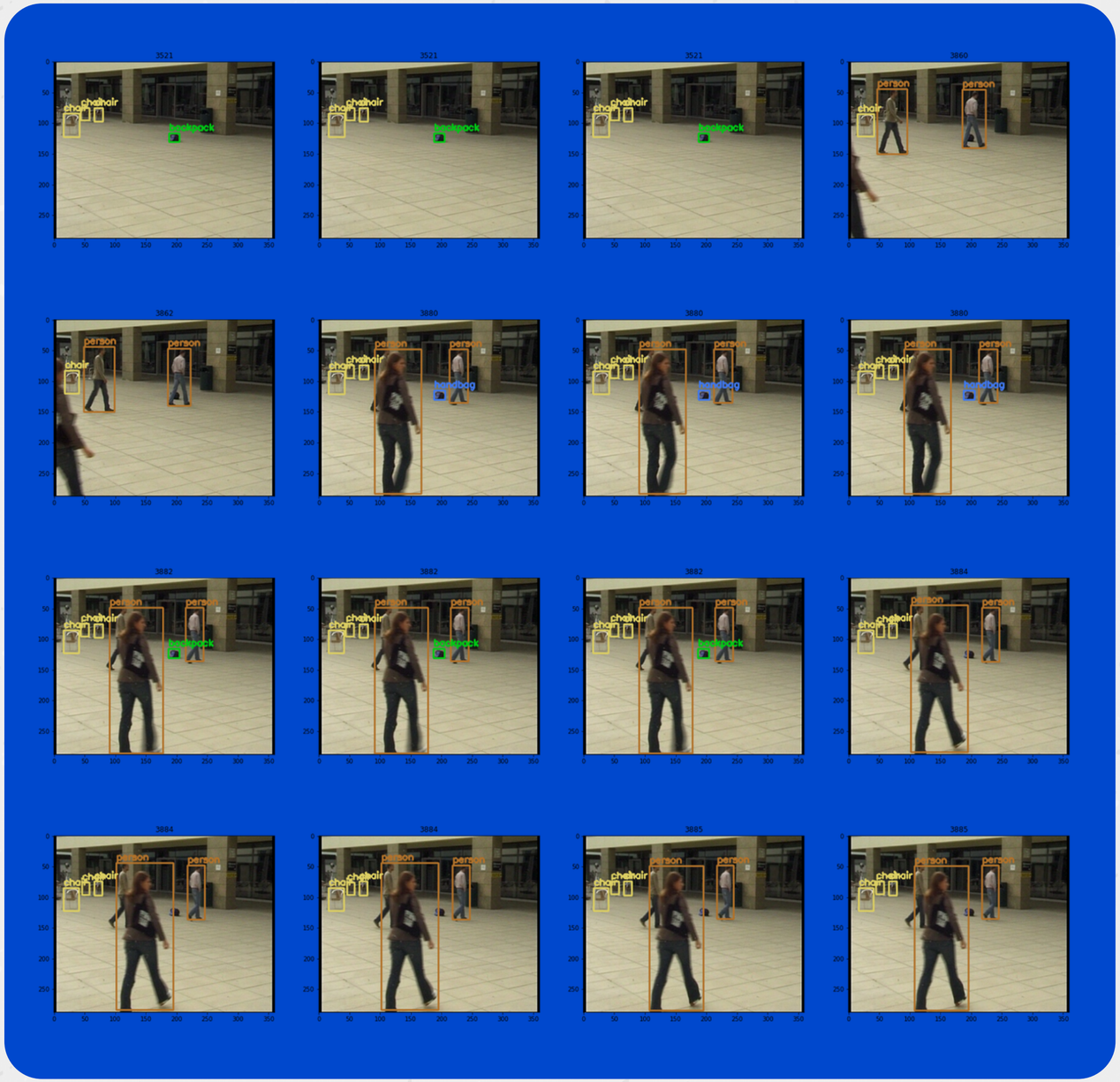
# Probable Technologies

- Flask/SQL
- AWS
- Tesseract

# Results

Precision and Speed
Solution-Part 2.2

- 16 color images took **3.4s**
- 16 Grayscale images took **0.45s**
- Can be drastically improved by storing the preprocessed images and just searching the database.



# Analysis

- For a dataset of ~1.5 GB scanning through for images took 0.25 seconds with a linear search, which is a very good benchmark
- Implementation of O(log n) based searching techniques could vastly improve search times.
- Part 1 of the solution is a robust model and completely scalable when certain enhancements are added to improve the search times
- Part 2 of the solution works well when the features the user wants are defined as classes instead of specifying the context in a natural language
- The search query time is approximately 4 seconds which could be drastically improved when DBMS systems like MySQL are used to make efficient retrievals.
- Including a natural language processing engine that could identify the context. With this could further move towards the aim of the problem by making it more user interactable

# Enhancements and Future Scope

- Using a language model which works accordingly with problem 2
- Using a dataset for loading as per the requirement. Example: For problem 2 a dataset that involved classes like hair color, skin color.
- Integrating ReID with problem 2. We wanted to do this, but the necessary libraries were written in MATLAB
- Using graph based techniques for data storage to extract images in a much efficient manner.

# Enhancements and Future Scope

- Using APIs to ease the usage of the application by connecting the cropped image and the video source directly to the model
- One of the scopes of application for this project is in surveillance and maintaining the order in a locality
- It is also essential in video based indexing where while just searching for a given text could take you to the exact point in the video concerned with the search out of hundreds of videos.
- Can be really useful in automation, which could simplify the process of editing videos by searching for a specific object concerned.

# Thank You