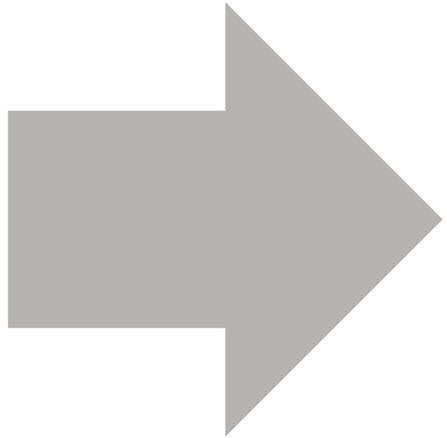
Active Learning

Recap

ThermalWear

Clothing Style (Type + Color)
+
Local Temperature
+
Local Humidity
=
People’s body temperature

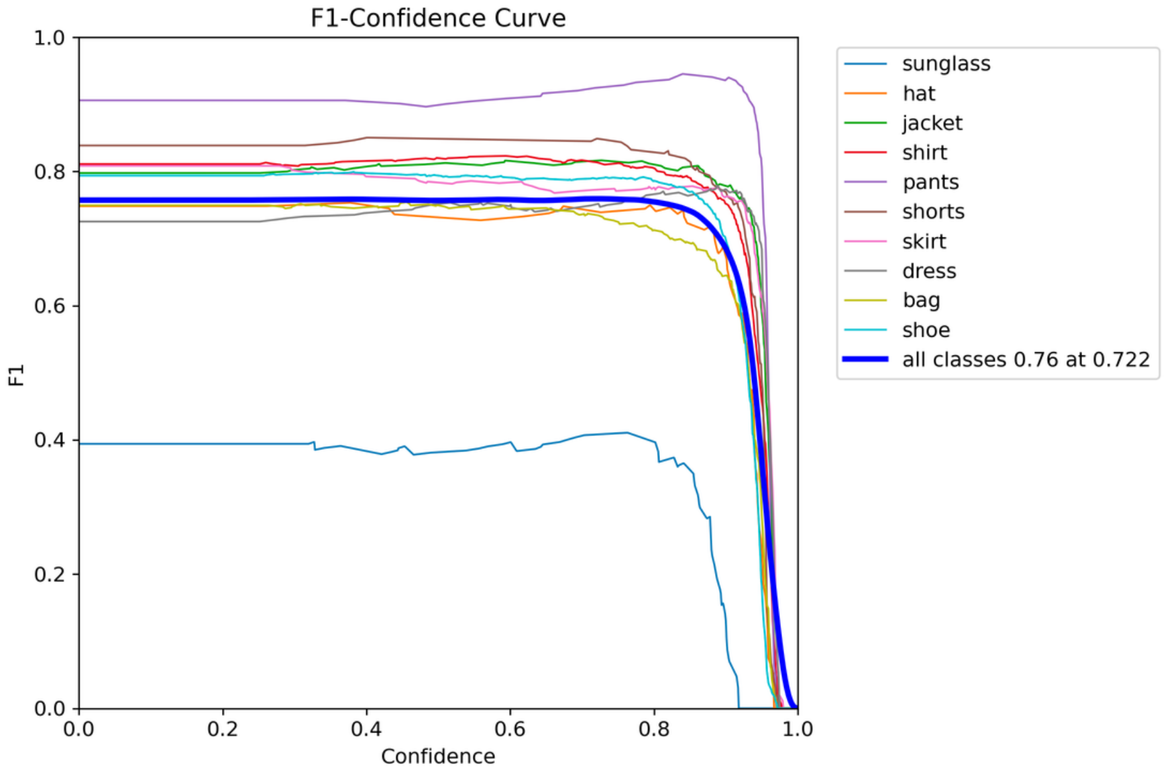


ComfyWear

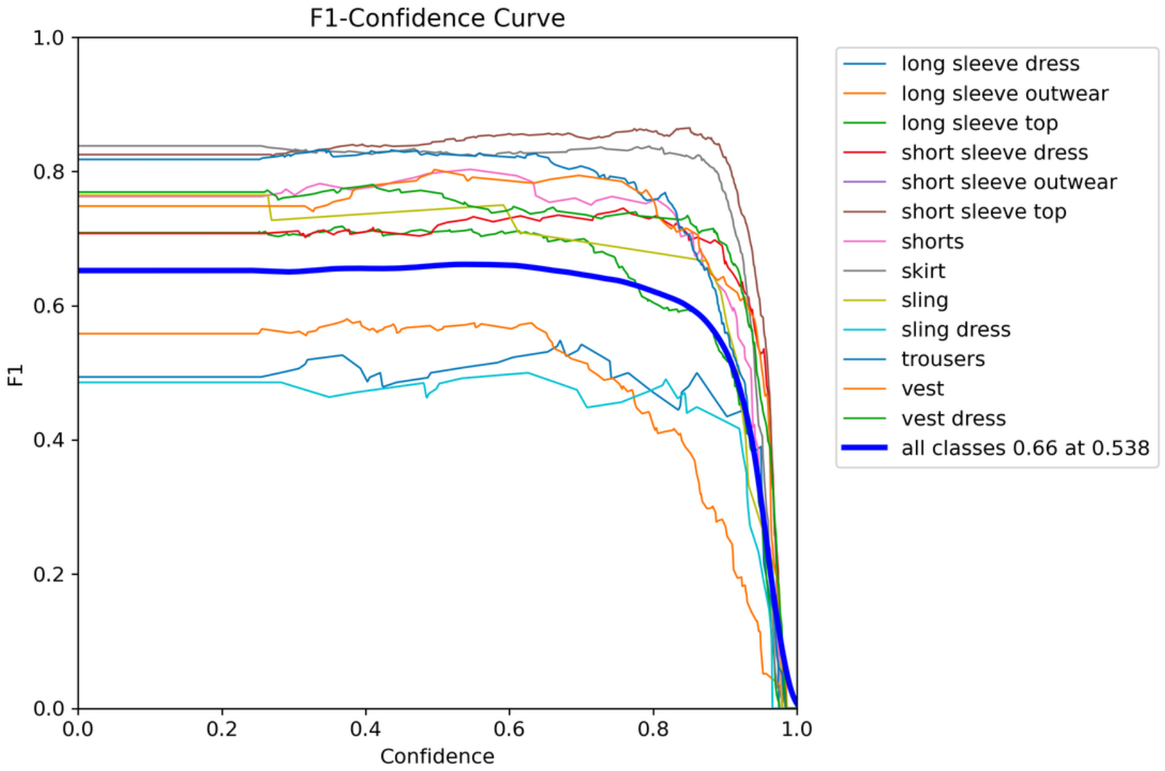
Clothing Style (Type only)
+
Local Temperature
+
Local Humidity
=
Comfort

F1-Confidence Curve

Colorful



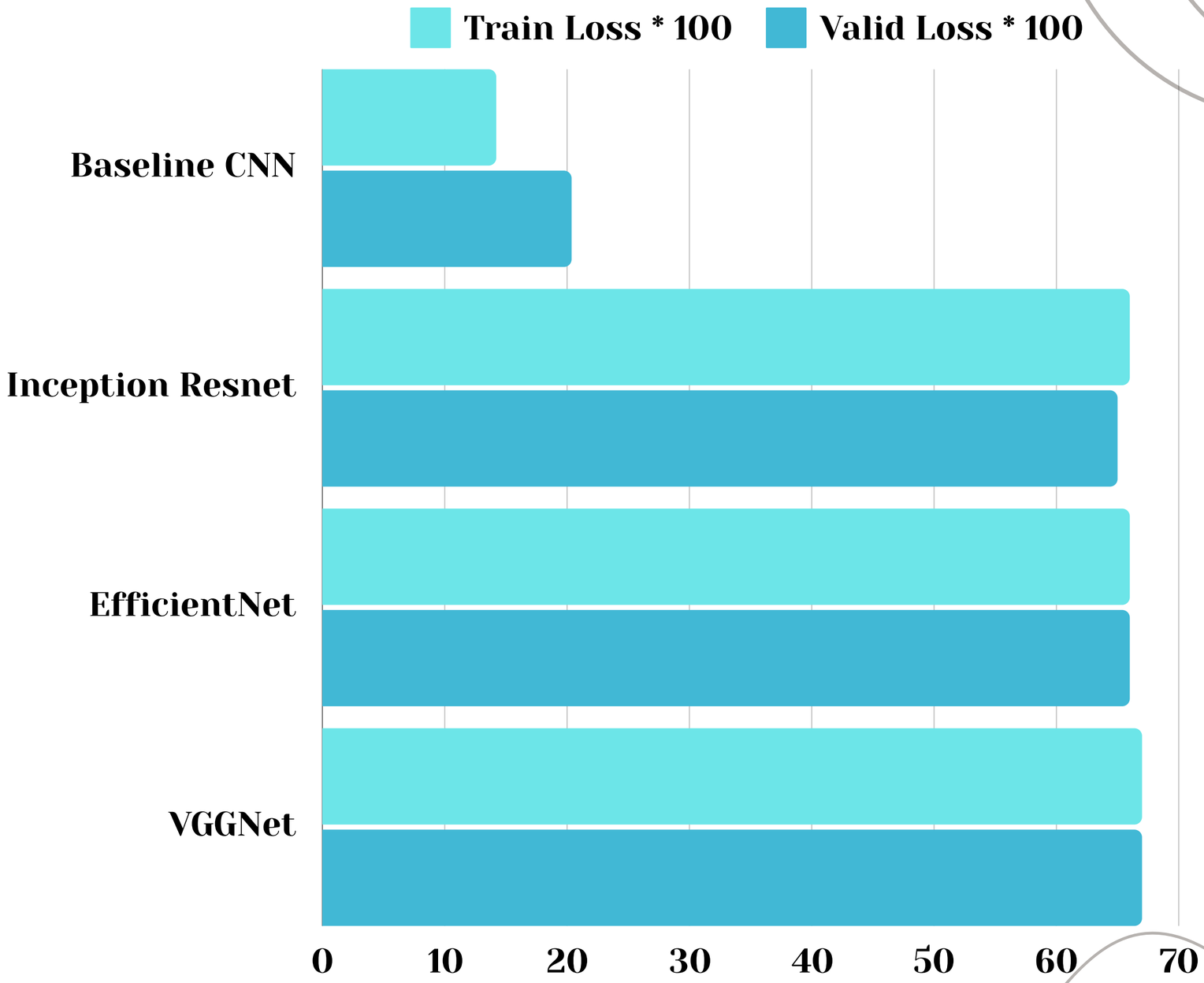
Deepfashion2



Soft-Biometric (UPAR)

20 epochs-MultiLabels

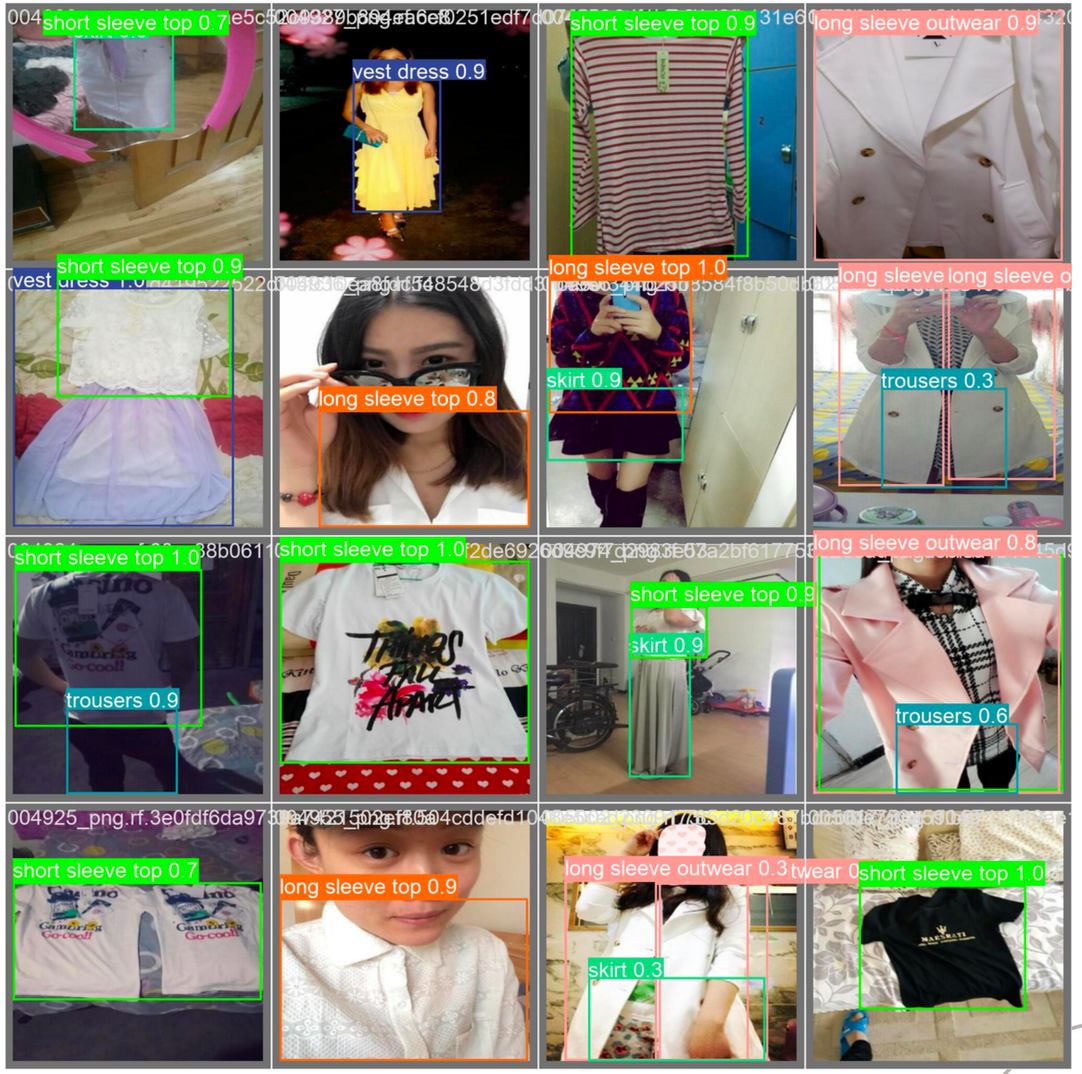
BCEWithLogitsLoss():
Sigmoid + BCELoss



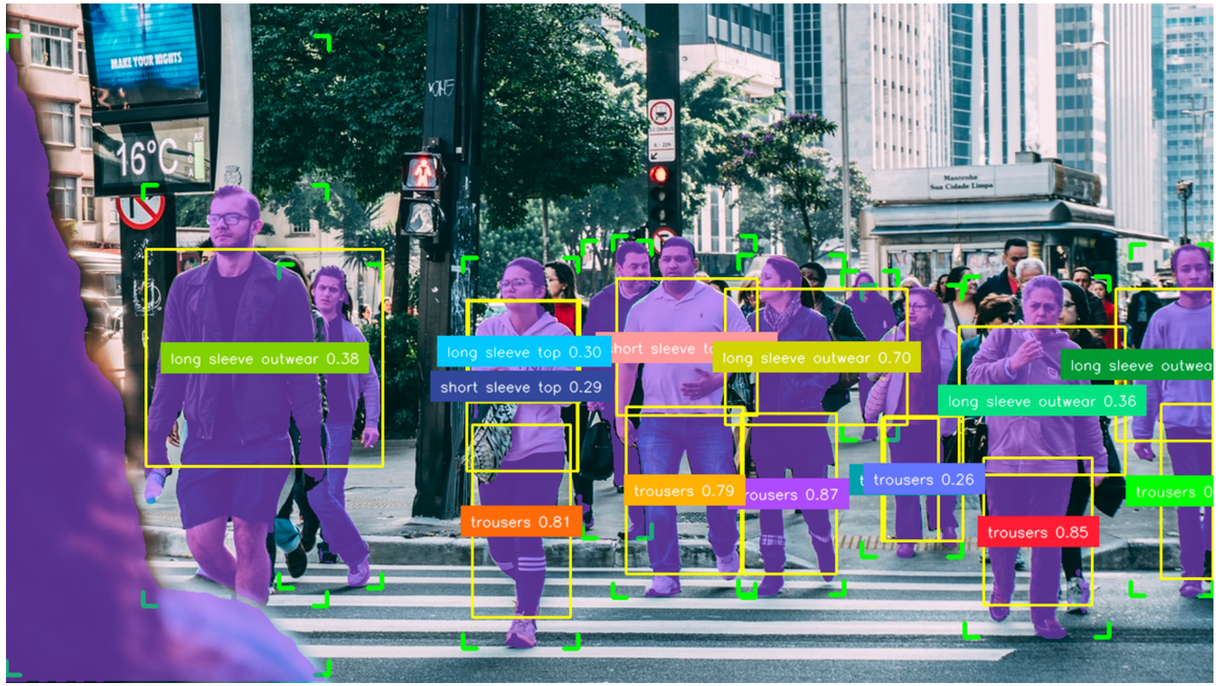
Example

Colorful

Deepfashion2

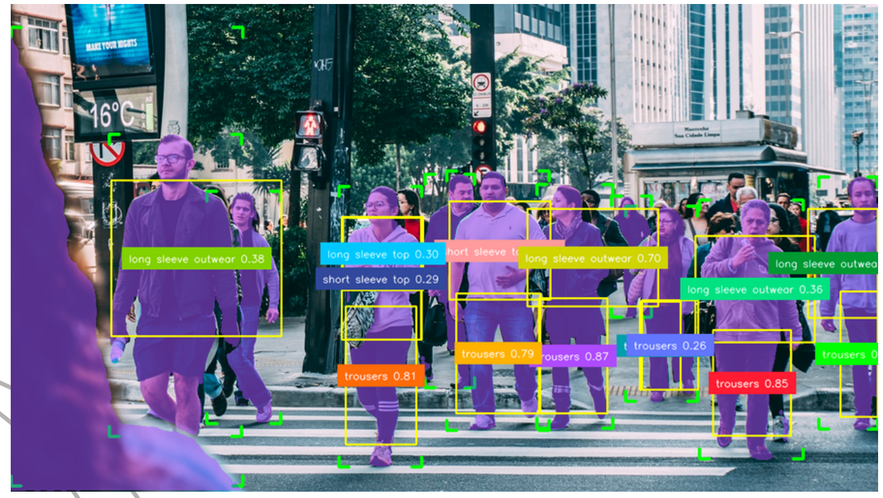



jacket 0.77
dress 0.62
shirt 0.73
shirt 0.88
pants 0.86
pants 0.80
shirt 0.29


long sleeve outwear 0.38
long sleeve top 0.30
short sleeve top 0.29
long sleeve outwear 0.70
long sleeve outwear 0.36
trousers 0.79
trousers 0.81
trousers 0.26
trousers 0.85

Number of Prediction

14



24

2 2 3 2 2

2 3 3 2 1 2

Feedback

- Collect 1-5 comfort level
- Detect the color of clothes
- People Re-Identification -> Create own dataset
- Human reinforcement

Today’s topic

Human Reinforcement

Reinforcement Learning

A machine learning paradigm that aims to train an agent (a software program or system) to make optimal decisions or take optimal actions in an environment to maximize a reward signal or cumulative reward over time

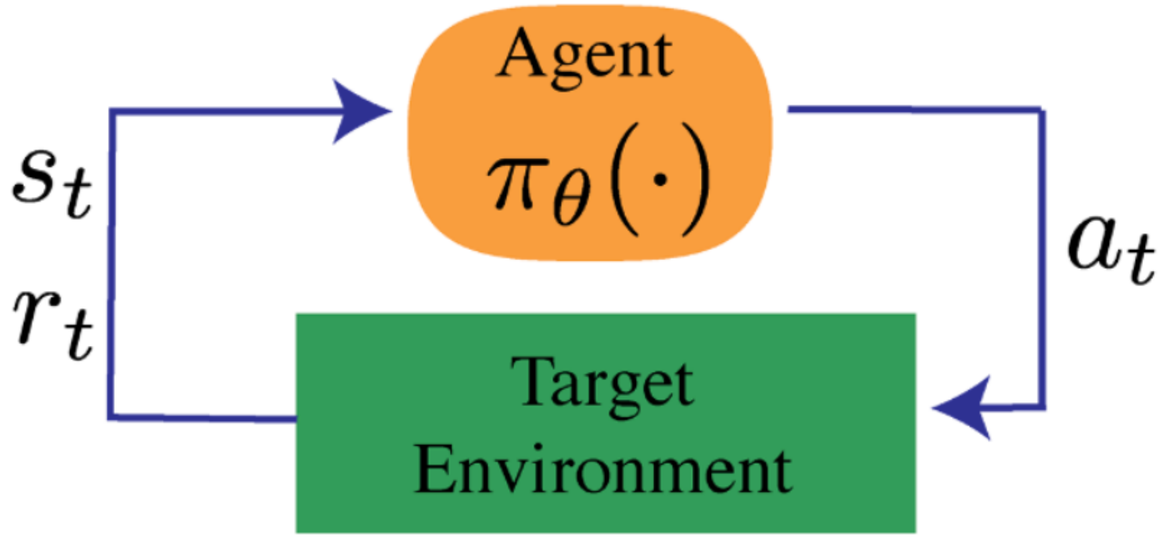
Key Elements

- **Agent**: The Learner
- **Environment**: The world or system in which the agent operates and takes actions.
- **State**: The current situation or condition of the environment that the agent perceives.

Key Elements

- **Action:** The choices or decisions that the agent can make to change its state in the environment.
- **Reward**: The feedback signal (positive or negative)
- **Policy**: A strategy or set of rules that an agent follows to make decisions.

How does RL work?



Some notation:

s_t : state
r_t : reward
a_t : action
$a_t \sim \pi_\theta(s_t)$: policy

RL vs SL?

Similarities

- Goal
- Learning from data
- Optimization

Differences

- Input/Output Mapping
  - State -> Action, X->Y
- Feedback Mechanism
- Exploration vs. Exploitation

Problems

How can we create a loss function for:

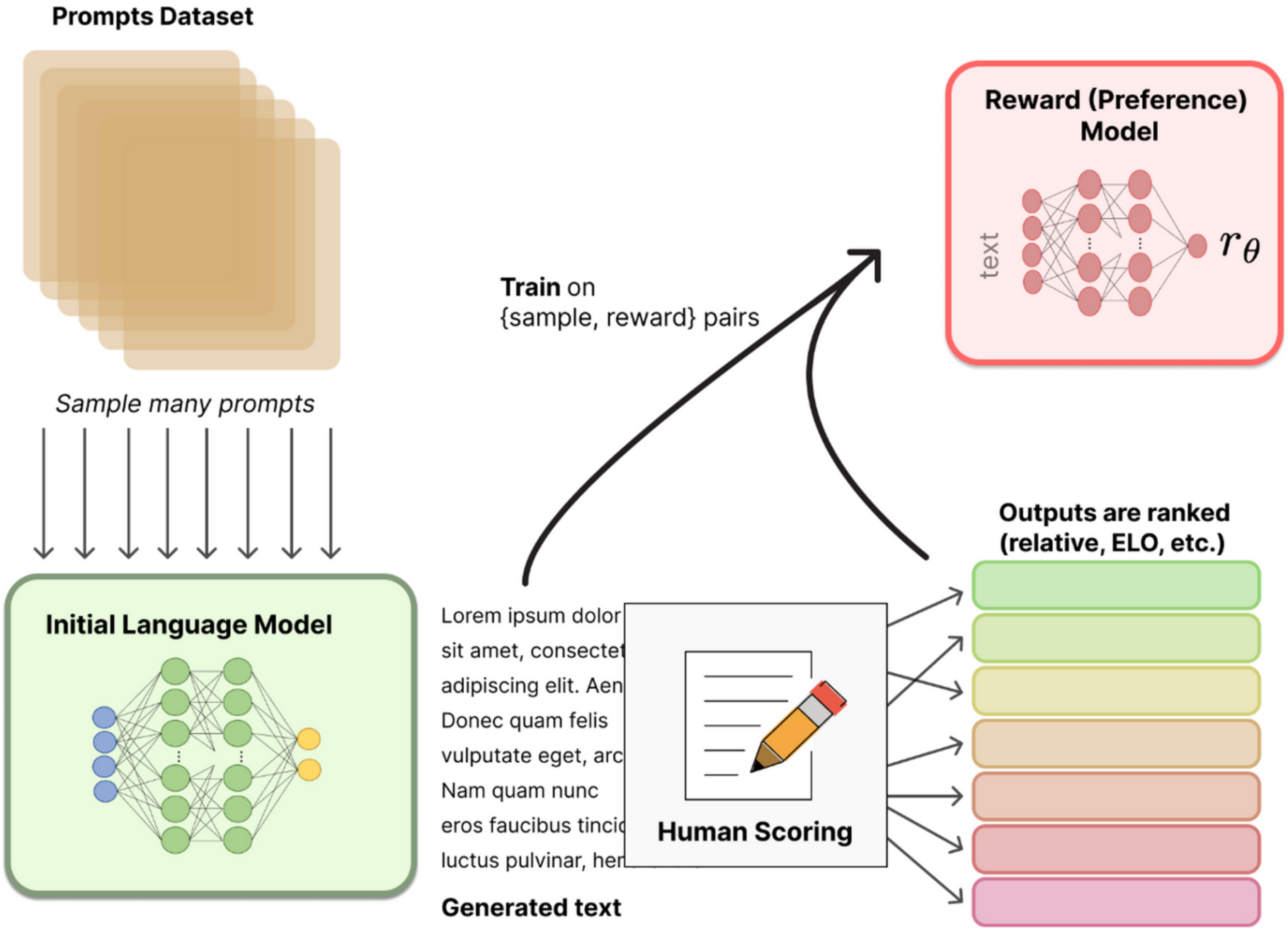
- What is funny?
- What is *ethical*?
- What is *safe*?
- What is taste?

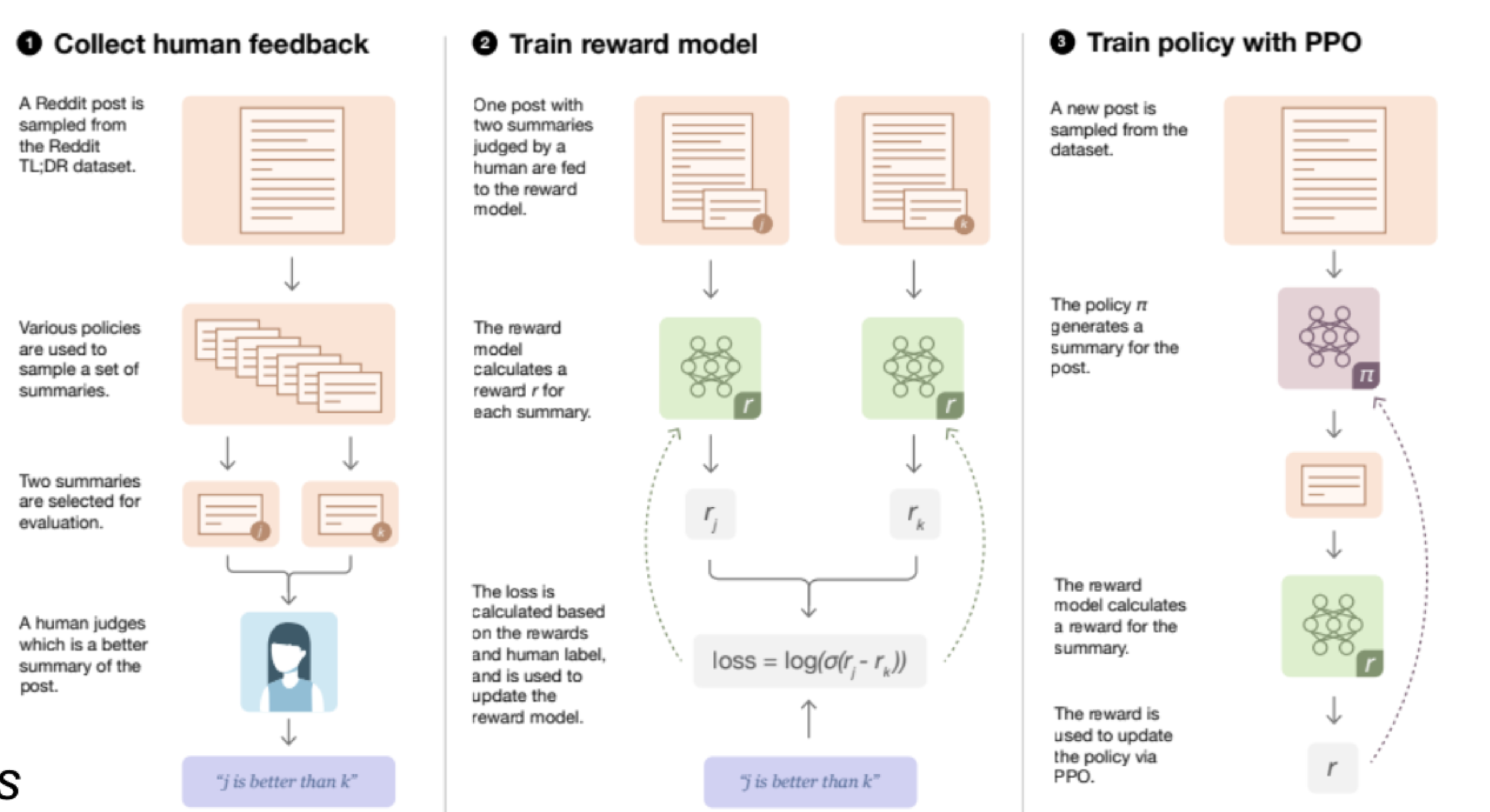
Reinforcement Learning Human Feedback (RLHF)

RLHF

The process of incorporating human feedback into machine learning algorithms to enhance learning efficiency and accuracy.

Mostly used in NLP tasks.


Prompts Dataset
Sample many prompts
Initial Language Model
Lorem ipsum dolor
sit amet, consectet
adipiscing elit. Aen
Donec quam felis
vulputate eget, arc
Nam quam nunc
eros faucibus tincid
luctus pulvinar, her
Generated text
Human Scoring
Train on
{sample, reward} pairs
Reward (Preference) Model
text
r_θ
Outputs are ranked
(relative, ELO, etc.)


1 Collect human feedback
A Reddit post is sampled from the Reddit TL;DR dataset.
Various policies are used to sample a set of summaries.
Two summaries are selected for evaluation.
A human judges which is a better summary of the post.
"j is better than k"
2 Train reward model
One post with two summaries judged by a human are fed to the reward model.
The reward model calculates a reward r for each summary.
The loss is calculated based on the rewards and human label, and is used to update the reward model.
r_j
r_k
loss = log(σ(r_j - r_k))
"j is better than k"
3 Train policy with PPO
A new post is sampled from the dataset.
The policy π generates a summary for the post.
The reward model calculates a reward for the summary.
The reward is used to update the policy via PPO.
r

Fine-Tuing RL

Kullback-Leibler Divergence

Prevent the model from deviating too much from the original pre-trained model.

Reward model

PPO

So should we use the RL?

CNN is better

So should we use the RL?

Reward Shaping Complexity

CNN is better

So should we use the RL?

Reward Shaping Complexity

CNN is better

Lack of Sequential Decision Making

So should we use the RL?

Reward Shaping Complexity

CNN is better

Lack of Sequential Decision Making

So should we use the RL?

Pretrained did well already

How to train LLM?

How to train LLM?

1. **Pre-training on Text Data**: Using a self-supervised objective like predicting masked words. (Unsupervised)
2. **Fine-tuning on Labeled Data:** The pre-trained LLM is then further trained on labeled datasets for specific tasks (Supervised)
3. **Getting Human Feedback**: Step 1 in RLHF
4. **Reinforcement Learning**: Step 2 in RLHF
5. **Creating New Datasets:** Use human to make high-quality output references
6. **Dataset Fine-tuning:** Train on dataset from 5
7. Repeat 3-6

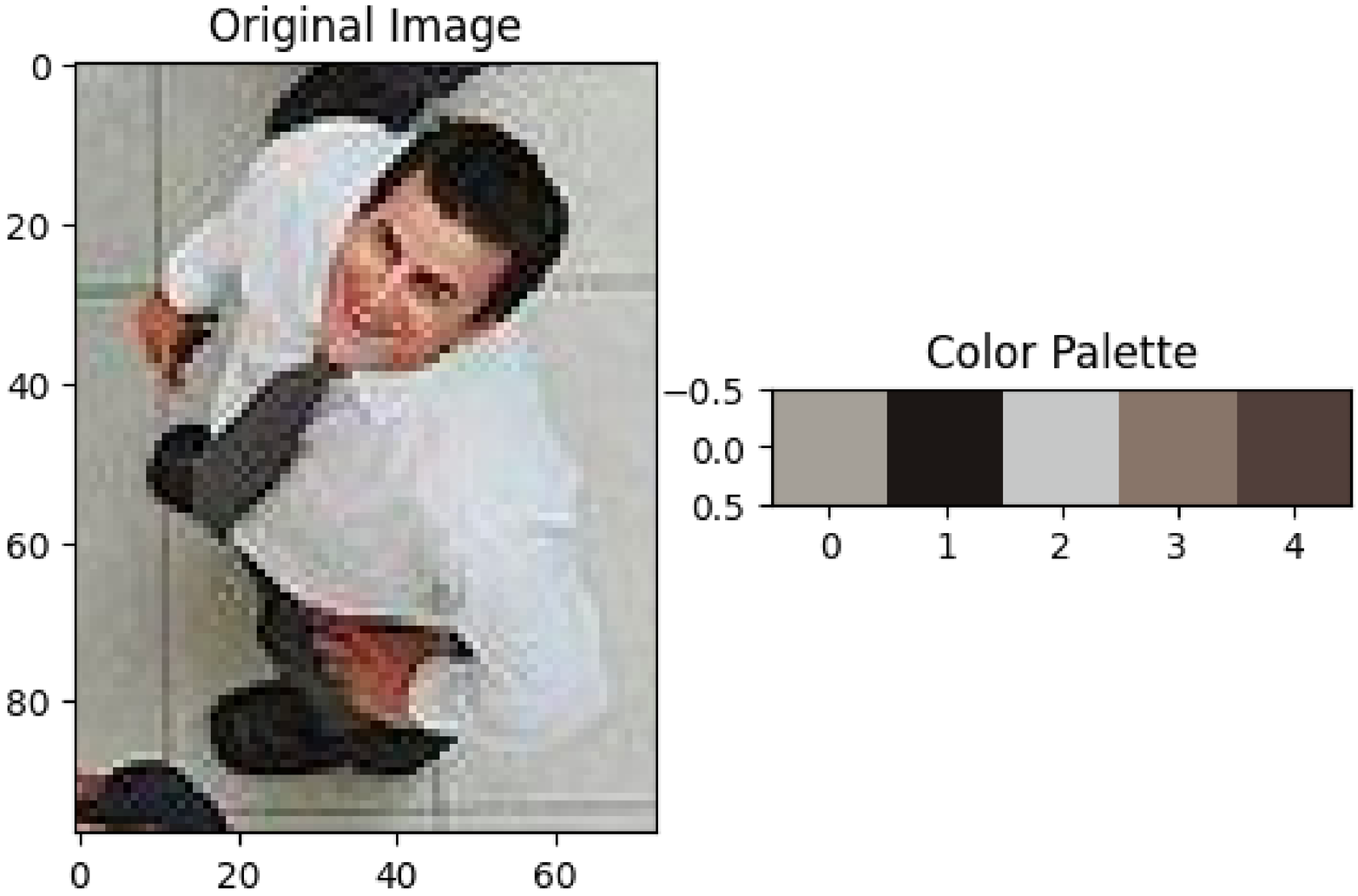
(Google's LaMDA, Anthropic's Constitutional AI, and OpenAI's InstructGPT)

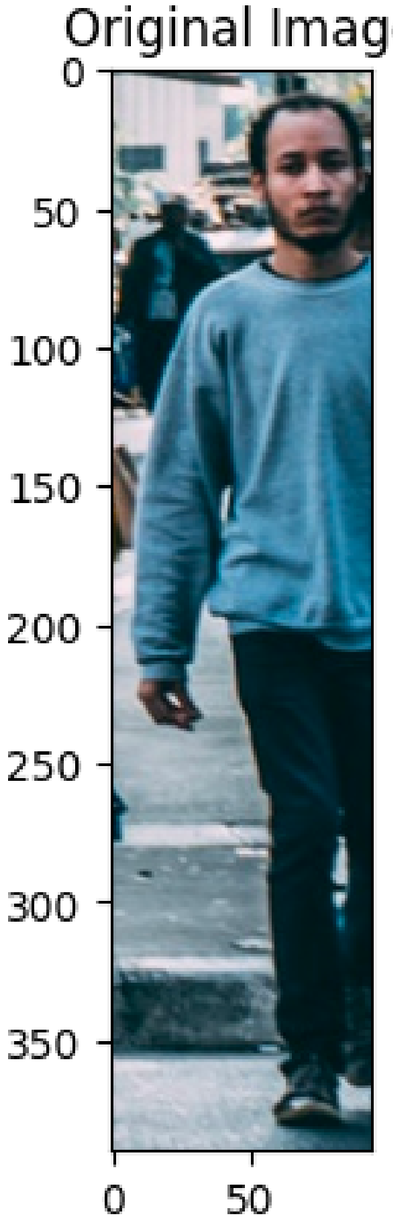
Extract The Color From detection

KMeans Clustering

```python
from sklearn.cluster import KMeans
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg

image = mpimg.imread('person/person/im.jpg4.jpg')
image = image / 255.0
w, h, d = tuple(image.shape)
pixel = np.reshape(image, (w * h, d))

ncolors = 5
model = KMeans(n_clusters=n_colors, random_state=42)
labels = model.fit_predict(pixel)
colors = model.cluster_centers
colors = (colors * 255).astype(np.uint8)

plt.subplot(1, 2, 1)
plt.imshow(image)
plt.title('Original Image')
```

- Normalize the image
- Flatten the pixel data into 2D
- Initialize k centroids (number of colors)
- Assign data points to clusters
- Update centroids
- Repeat


Original Image
0
20
40
60
80
0
20
40
60
Color Palette
−0.5
0.0
0.5
0
1
2
3
4


Original Image
0
50
100
150
200
250
300
350
0
50

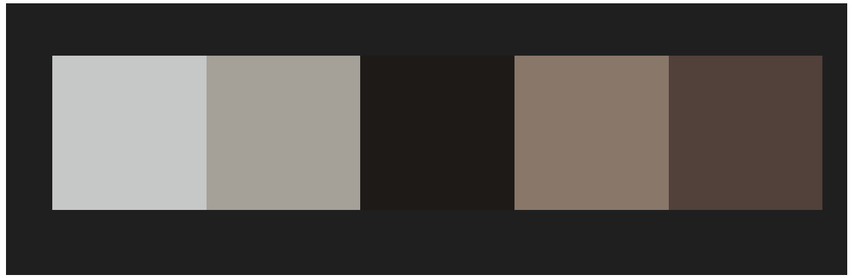

Color Palette
−0.5
0.0
0.5
0
1
2
3
4

Pylette

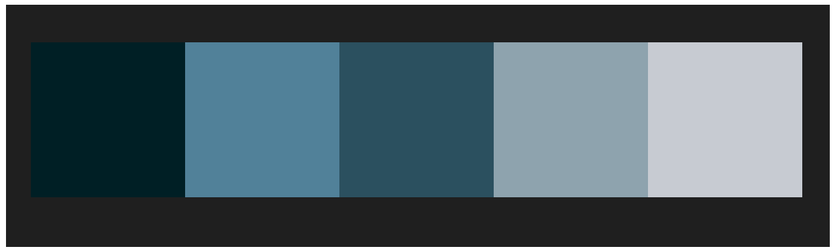
```python
from Pylette import extract_colors
palette = extract_colors(image='person/person/im.jpg4.jpg', palette_size=5, resize=True)
```

- Pylette is optimized for color extraction and palette generation, making it more specialized and potentially more accurate for this task compared to the general-purpose k-means algorithm.
- Uses a perceptual color clustering algorithm based on the CIEDE2000 color difference formula


Original Image
0
20
40
60
80
0
20
40
60

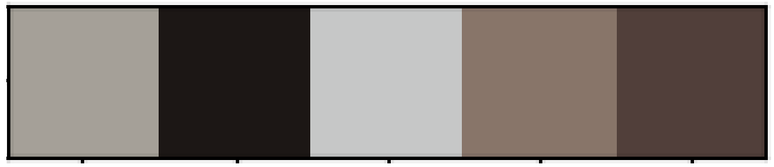



Original Image
0
50
100
150
200
250
300
350
0
50



Compare the result

KMeans



Pylette

Compare the result

KMeans

Pylette

Problem

Color Mapping

Our Label

- (255, 0, 0) -> Red
- (255, 165, 0) -> Orange
- (255, 255, 0) -> Yellow
- (0, 128, 0) -> Green
- (0, 0, 255) -> Blue
- (75, 0, 130) -> indigo
- (238, 130, 238) -> Violet
- White, Black

What we extracted

- (255, 102, 102)
- (255, 204, 102)
- (255, 255, 102)
- (102, 204, 102)
- Doesn’t matched any label

So, how can we solve that?

Map the RGB to our closest label

Manually map by distance

rgb

```
def euclidean_distance(color1, color2):
    return sqrt(sum((c1 - c2) ** 2 for c1, c2 in zip(color1, color2)))
```

Manually map by distance

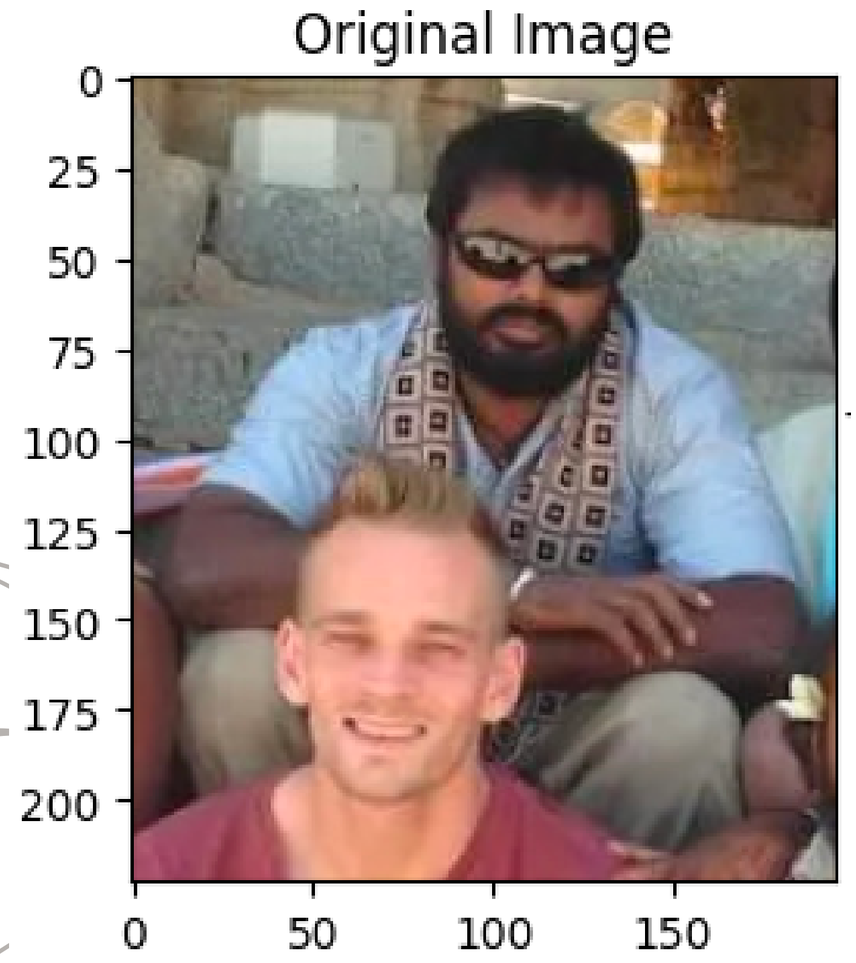
rgb

```
def euclidean_distance(color1, color2):
    return sqrt(sum((c1 - c2) ** 2 for c1, c2 in zip(color1, color2)))
```

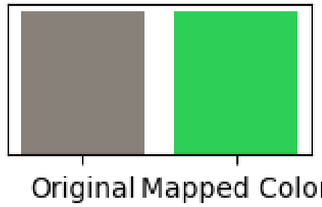
Might not really accurate

```python
def map_to_rainbow_color(rgb):
    rainbow_colors = {
        'red': (255, 0, 0),
        'light_red': (255, 102, 102),
        'dark_red': (153, 0, 0),
        'orange': (255, 165, 0),
        'light_orange': (255, 204, 102),
        'dark_orange': (204, 102, 0),
        'yellow': (255, 255, 0),
        'light_yellow': (255, 255, 102),
        'dark_yellow': (204, 204, 0),
        'green': (0, 128, 0),
        'light_green': (102, 204, 102),
        'dark_green': (0, 102, 0),
        'blue': (0, 0, 255),
        'light_blue': (102, 102, 255),
        'dark_blue': (0, 0, 153),
        'indigo': (75, 0, 130),
        'light_indigo': (138, 43, 226),
        'dark_indigo': (49, 0, 98),
        'violet': (238, 130, 238),
        'light_violet': (255, 187, 255),
        'dark_violet': (147, 112, 219),
        'black': (0, 0, 0),
        'white': (255, 255, 255)

    }
    closest_color = min(rainbow_colors, key=lambda color: euclidean_distance(rgb, rainbow_colors[color]))
    return closest_color, rainbow_colors[closest_color]
```

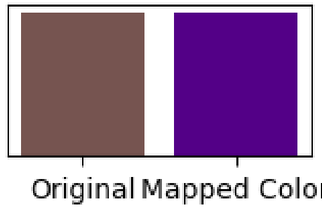
Result



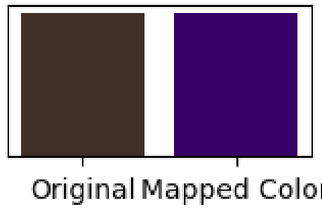

```
RGB: (135, 129, 123) -> Mapped Color: light_green
```



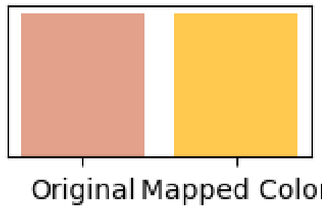

```
RGB: (113, 85, 81) -> Mapped Color: indigo
```



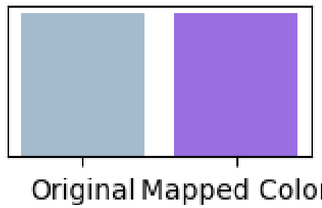

```
RGB: (62, 48, 43) -> Mapped Color: dark_indigo
```




```
RGB: (217, 163, 142) -> Mapped Color: light_orange
```




```
RGB: (168, 186, 203) -> Mapped Color: dark_violet
```

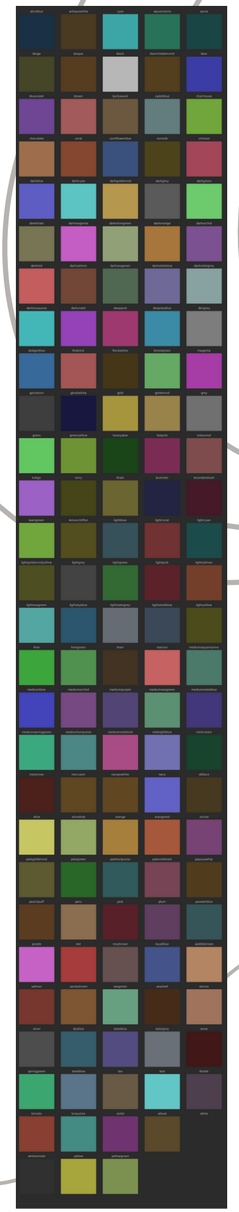


The result isn't good.
It might be because of a lack of mapping rgb.

Webcolors

Webcolors

```python
from webcolors import (
    CSS3_HEX_TO_NAMES,
    hex_to_rgb,
)
```



138 colors

Webcolors

```
from scipy.spatial import KDTree
```

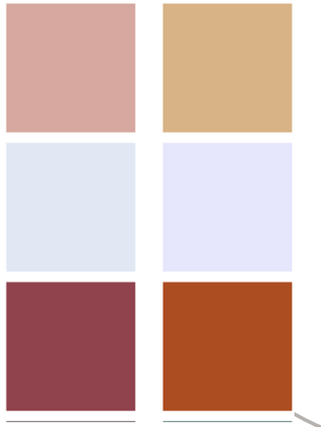
```
def convert_rgb_to_names(rgb_tuple):
    css3_db = css3_hex_to_names
    names = []
    rgb_values = []
    for color_hex, color_name in css3_db.items():
        names.append(color_name)
        rgb_values.append(hex_to_rgb(color_hex))

    kdt_db = KDTree(rgb_values)
    distance, index = kdt_db.query(rgb_tuple)
    return f'closest match: {names[index]}'
```

Performs a nearest neighbor search in the K-D Tree to find the closest RGB value to the given rgb_tuple.

Using Euclidean distance

RESULTS

RESULTS



UPAR

Compare Models

Resnet50

- Same model as first place in the UPAR Challenge Competition.
- Trained 50 epochs
- ~37% accuracy

MobileNet

- Trained 50 epochs
- ~95% accuracy
- Haven’t evaluate yet

Person Re-Indentification

A Comprehensive Overview of Person Re-Identification Approaches (2020)

Steps

j

1. **Local information extraction**: Extract feature from diffrent body blocks/parts.
2. **Global information extraction**: Extract feature from multiple non-overlapped cameras.
3. **Metric Learning:** Find an optimized distance metric, designing new distance functions and loss functions.

Steps

4. **Post processing**: Applying re-ranking algorithms on the initial ranking list obtained from the re-identification model.

Steps

4. **Efficiency improvement**: All about the models.

5. **Labeling cost reduction**: Data augmentation, unsupervised domain adaptation, semi-supervised learning. (reduce the amount of labeled training data required).

6. **Data type extension**: Images -> videos, depth images, infrared images, low-resolution images.

A Comprehensive Overview of Person Re-Identification Approaches (2020)

Steps

j

1. **3D CNN**: Allows the network to learn spatial and temporal features jointly (Clip-level instead of Frame-level)
2. **Split video into short clips**: The video is too long.
3. **Pair each clip**: MxM -> M is a number of clips.
4. **Compute importance for each pair**: Using MLP network. This score indicates how informative that clip pair is for estimating video similarity

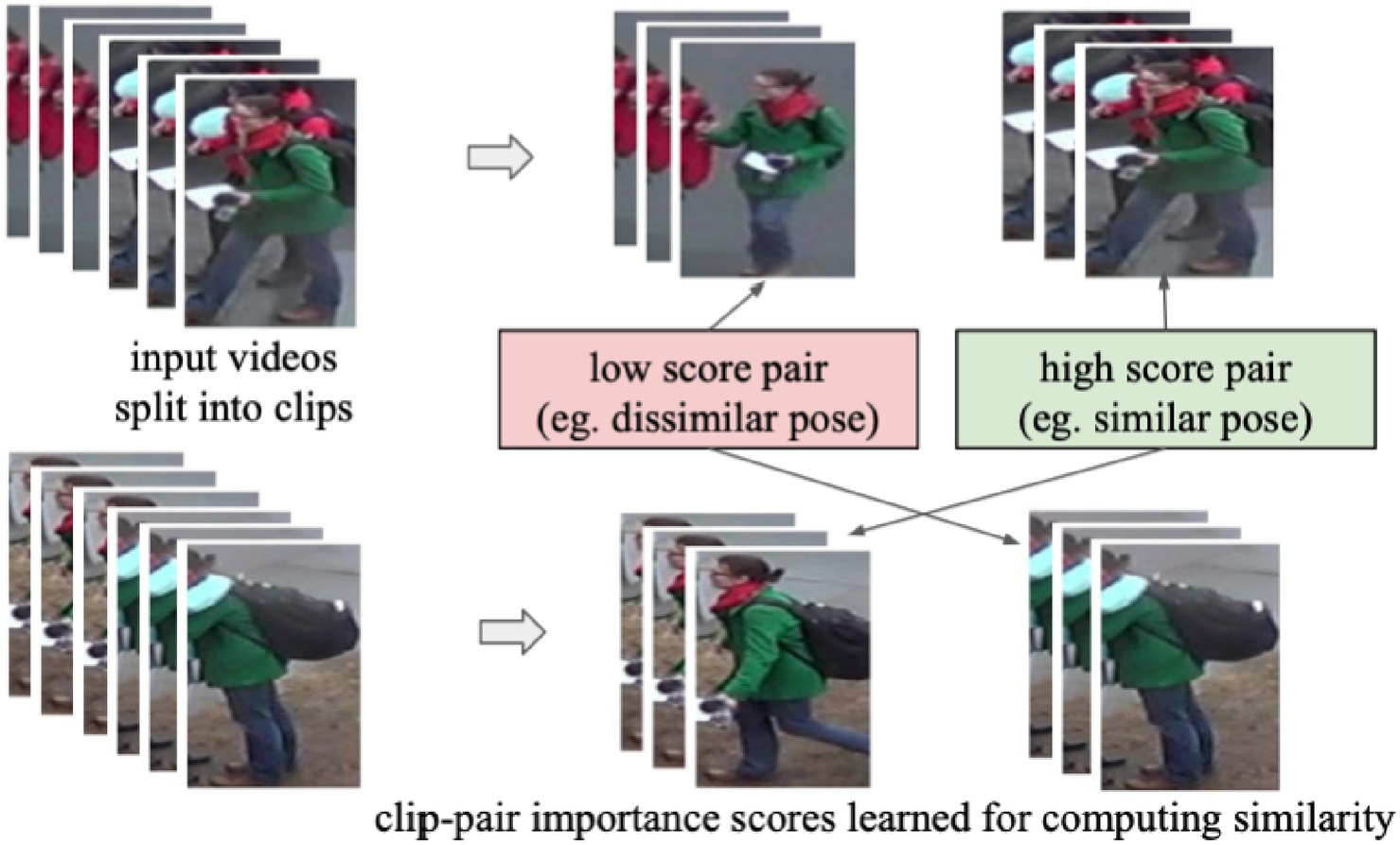
Steps

j

5. **Train the model**: The model learns to focus on the most informative clip pairs for re-identification, while automatically downweighting the contribution of noisy or uninformative clips.

(Weighted Average)

Steps


input videos
split into clips
low score pair
(eg. dissimilar pose)
high score pair
(eg. similar pose)
clip-pair importance scores learned for computing similarity

Unsupervised Pre-training for Person Re-identification (2021)

Steps

j

1. **LUPerson**: The authors build a large-scale unlabeled person Re-ID dataset called "LUPerson" containing over 4 million images of 200K identities.
2. **MoCoV2**: Using contrastive learning (One of the self-supervised learning)
3. **Direct applying MoCoV2 doen’t work**: Data augmentation, tuning the model parameters.

ByteTrack

ByteTrack

One of the algorithms for tracking the object in the video.

How does ByteTrack work?

- **Object detection**: Use YoloX object detection model as its backbone (YoloX is a single-stage object detector)
- **Feature Extraction**
- **Compute association score:** Computes an association score between each detected object in the current frame and the tracked objects from the previous frames by considers the spatial proximity (IoU) and appearance similarity (cosine distance) between the objects.

How does ByteTrack work?

- **A hierarchical association:** Handle occlusions and missing detections.
- **Track management**:
  - Maintain active track (Unique object)
  - When detect new object -> Assign a new track
  - When object dissappear for a certain time ->
    - Remove from a track

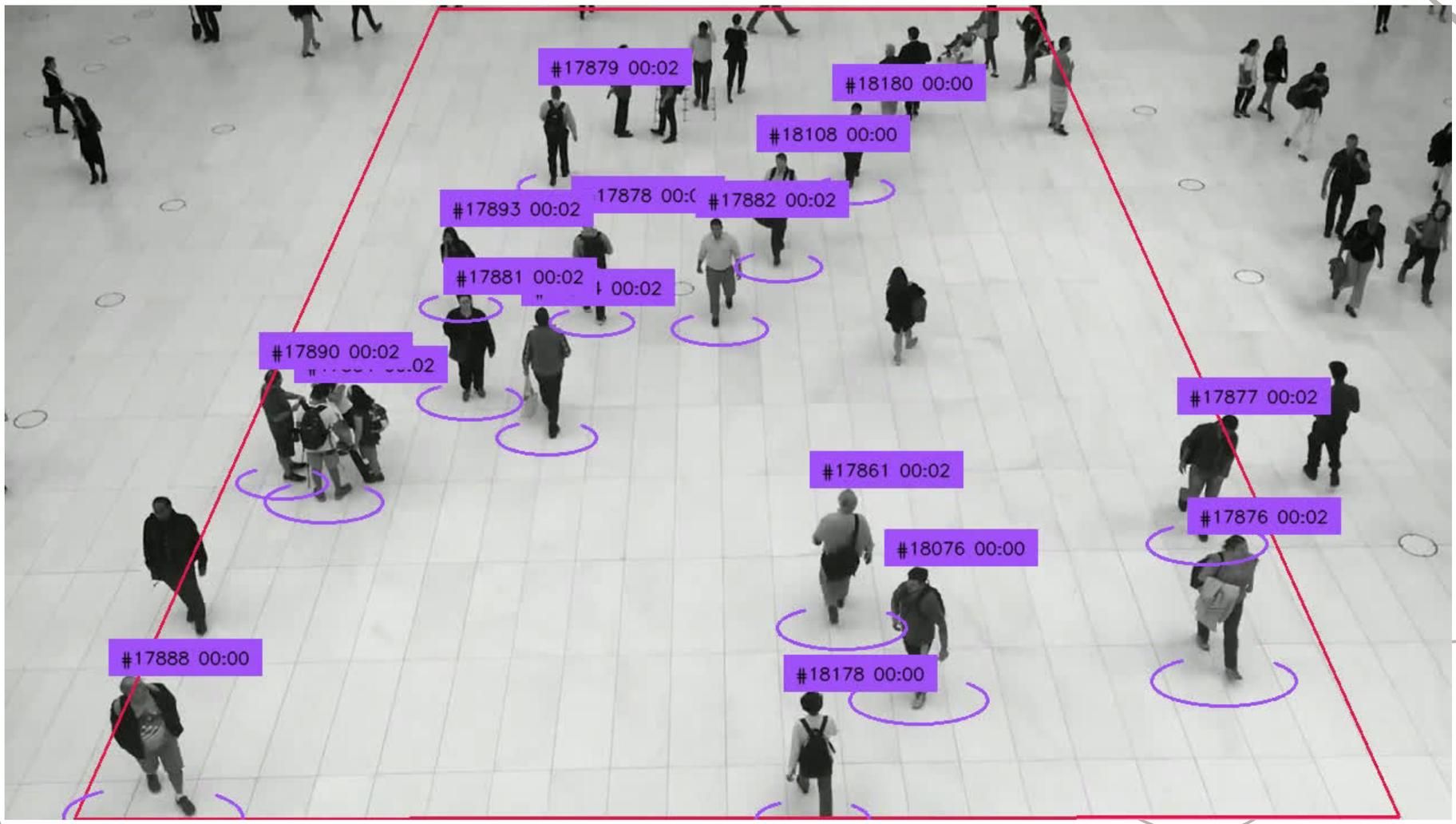
How does ByteTrack work?

- **Motion Model**
  - Employs a simple motion model to predict the position of tracked objects in the next frame.
  - The motion model assumes a constant velocity and updates the object's position based on its previous location and velocity.

Implementation

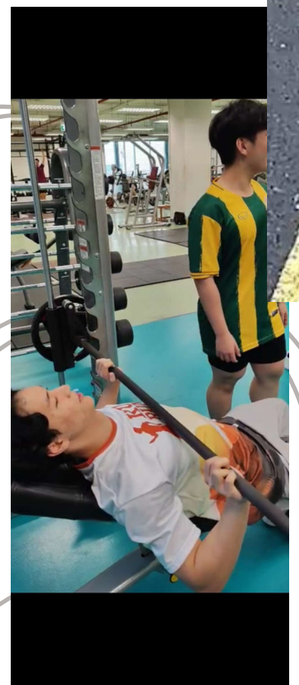
Implementation

Implementation

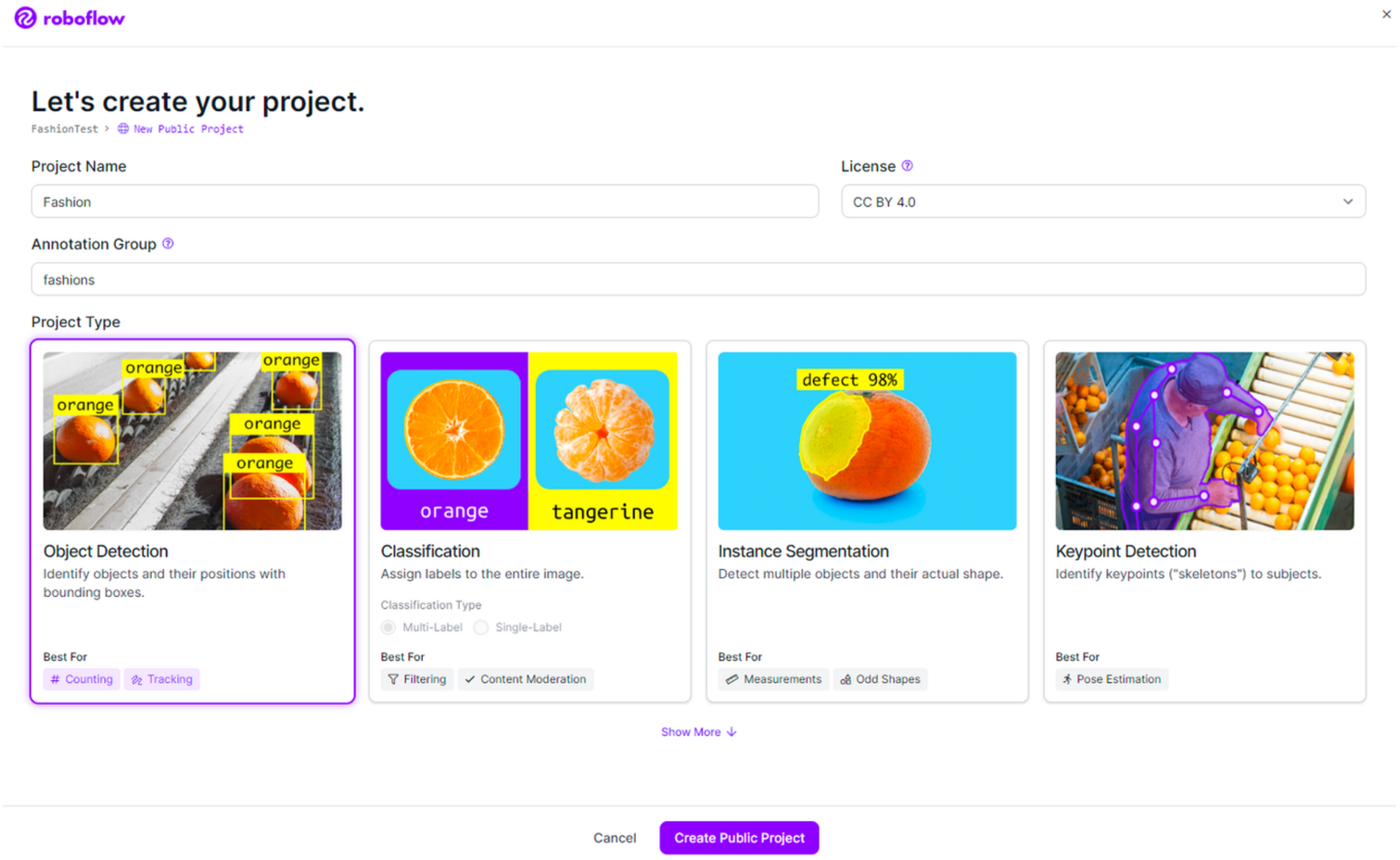


Active Learning

Active Learning



Create a project



Upload to Roboflow

```
dir_name = f"{HOME}/yolov9/mytest"
file_extension_type = [".jpg", ".png", ".jpeg", ".webp"]

image_glob = [file for ext_type in file_extension_type for file in glob.glob(dir_name + '/*' + ext_type)]

for image_path in image_glob:
    annot_path = image_path.rsplit(".", 1)[0] + ".txt"
    annot_path = annot_path if os.path.exists(annot_path) else None
    print(project.single_upload(
        image_path=image_path,
        annotation_path=annot_path,
    ))
```

```
{'image': {'success': True, 'id': 'u1WIOKPk1gN7YBcDVloF'}, 'annotation': {'success': True}, 'upload_time': 5.829010963439941, 'annotation_time': 1.00199
19872283936}
{'image': {'success': True, 'id': 'drrteocMALyStRLbKZcC'}, 'annotation': {'success': True}, 'upload_time': 4.348002672195435, 'annotation_time': 1.12300
0144958496}
{'image': {'success': True, 'id': 'HOXR5hYMFx48V4W7XZt6'}, 'annotation': {'success': True}, 'upload_time': 2.7090094089508057, 'annotation_time': 0.8689
875602722168}
{'image': {'success': True, 'id': 'qRrpojSG9zBcsLgBQDMJ'}, 'annotation': None, 'upload_time': 2.6800103187561035, 'annotation_time': None}
{'image': {'success': True, 'id': 'HIBtCUxRyPYJO406cXTt'}, 'annotation': {'success': True}, 'upload_time': 0.9860019683837891, 'annotation_time': 0.9089
875221252441}
{'image': {'success': True, 'id': 'XFoOtpl1lUmgWsBkit6E'}, 'annotation': {'success': True}, 'upload_time': 1.8720018863677979, 'annotation_time': 0.8840
103149414062}
{'image': {'success': True, 'id': 'KZpD2s4fjOaAJY1ja10U'}, 'annotation': {'success': True, 'warnings': True, 'badAnnotations': [{'type': 'warning', 'cod
e': 'annotation:trimmed', 'key': 'Screenshot_20240408_123727_Gallery.jpg', 'label': '10'}, {'type': 'warning', 'code': 'annotation:trimmed', 'key': 'Scr
eenshot_20240408_123727_Gallery.jpg', 'label': '10'}]}, 'upload_time': 2.6599998474121094, 'annotation_time': 0.8830046653747559}
{'image': {'success': True, 'id': 'HCStD9fOMLBd1CKutsde'}, 'annotation': {'success': True}, 'upload_time': 2.871997356414795, 'annotation_time': 0.86598
68240356445}
{'image': {'success': True, 'id': 'XxqxLdmT8P2BeV70zARY'}, 'annotation': {'success': True}, 'upload_time': 3.119680643081665, 'annotation_time': 0.91100
23975372314}
{'image': {'success': True, 'id': 'CEUEBttQAOgQyLgKqwYI'}, 'annotation': {'success': True}, 'upload_time': 2.4799888134002686, 'annotation_time': 1.3139
99891281128}
```

Demo