Closing Meeting

LOUIS-MELCHIOR GIRAUD - LOUIS MARTYR - JOALIE CORNELIE - EMRYS MEZIANI - RAPHAËL MICHON - MALO LEROY

Equipe 283

Lundi 18 mars 2024

Project Background : Step 1

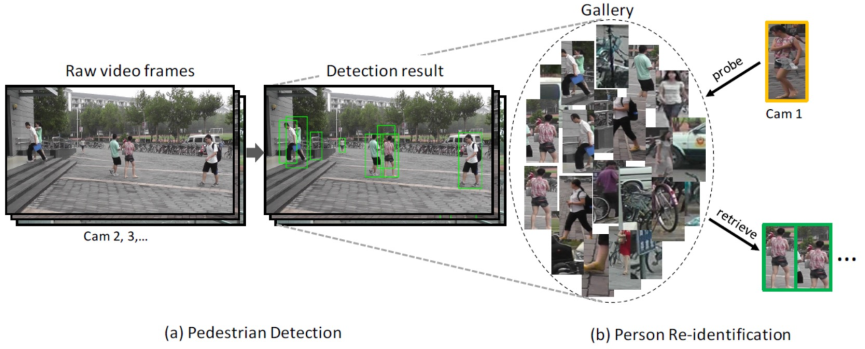
- Literature review on the subject of person re-identification
  - Main state-of-the-art re-identification models
  - Datasets used
  - Metrics for model evaluation
  - Writing a summary of our research

Project Background : Step 2

Preprocessing and Data Visualization

Creation of pytorch structures for training

PRW (Person Re-identification in the Wild) Dataset



CAMPUS Dataset

This dataset is collected and annotated by Yuanlu Xu, Hang Qi, Yang Liu, Yansong Tang and Nawin Waree. We are from Center for Vision, Cognition, Learning, and Autonomy (VCLA), University of California, Los Angeles (UCLA).

Project Background : Step 3

- Development of convolutional neural network (CNN) models for person re-identification
- Implementation of training/validation cycles:
- Evaluation of models on the test set
- Implementation of a tracker using the weights of the constructed model

Results obtained : Step 1

Knowledge

Deep learning systems

What is a CNN and different models

Distance metrics

Evaluation methods

Write a summary report of a survey

JOURNAL OF LaTeX CLASS FILES, VOL. 14, NO. 8, AUGUST 2015

Person Re-identification:
Past, Present and Future

Liang Zheng, Yi Yang, and Alexander G. Hauptmann

Abstract—Person re-identification (re-ID) has become increasingly popular in the community due to its application and research significance. It aims at spotting a person of interest in other cameras. In the early days, hand-crafted algorithms and small-scale evaluation were predominantly reported. Recent years have witnessed the emergence of large-scale datasets and deep learning systems which make use of large data volumes. Considering different tasks, we classify most current re-ID methods into two classes, *i.e.*, image-based and video-based; in both tasks, hand-crafted and deep learning systems will be reviewed. Moreover, two new re-ID tasks which are much closer to real-world applications are described and discussed, *i.e.*, end-to-end re-ID and fast re-ID in very large galleries. This paper: 1) introduces the history of person re-ID and its relationship with image classification and instance retrieval; 2) surveys a broad selection of the hand-crafted systems and the large-scale methods in both image- and video-based re-ID; 3) describes critical future directions in end-to-end re-ID and fast retrieval in large galleries; and 4) finally briefs some important yet under-developed issues.

Index Terms—Large-scale person re-identification, hand-crafted systems, Convolutional Neural Network, literature survey.

1 INTRODUCTION

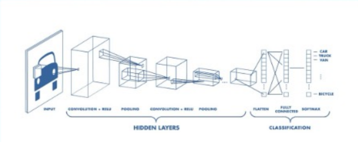
ACCORDING to Homer (*Odyssey* iv:412), *Mennelaus* was becalmed on his journey home from the Trojan War; He wanted to propitiate the gods and return safely home. He was told that he should capture *Proteus* and force him to

networks in theme parks, university campuses and streets, *etc*. Both causes make it extremely expensive to rely solely on brute-force human labor to accurately and efficiently spot a person-of-interest or to track a person across cameras.

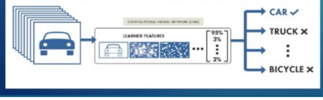
Person re-identification: Past, present and future,Zheng, Liang and Yang, Yi and Hauptmann, Alexander G. [en ligne][Consulté le 16/04/2024]

Image based Re-Id & Deep Learning model : CNN

Architecture d'un CNN



Output d'un CNN



Couche de convolution (CONV) :
Analyse les images fournies en entrée
Détecte la présence d'un ensemble de features.

Couche de Pooling (POOL) :
Réduire la taille des images.
Ex: max-pooling,average pooling.

Couche d'activation ReLU :
Remplace toutes les valeurs négatives reçues en entrées par des zéros.
Objectif: rendre le modèle non linéaire et de ce fait plus complexe.

Couche Fully Connected (FC) :
Classification de l'input image.
Envoie enfin en sortie un vecteur représentant la probabilité pour l'input image d'appartenir à une classe.

Rapport 1er Jalon

Introduction :

La réidentification de personnes est un domaine en pleine expansion, dont l'objectif est de reconnaître un individu à travers différentes caméras ou à différents moments. Dans le cadre de ce rapport, nous nous penchons sur les méthodes de réidentification présentées dans le survey qui nous a été attribué [1]. Nous explorerons les principales techniques, leurs caractéristiques, leurs avantages et inconvénients, ainsi que les datasets couramment utilisés dans ce domaine.

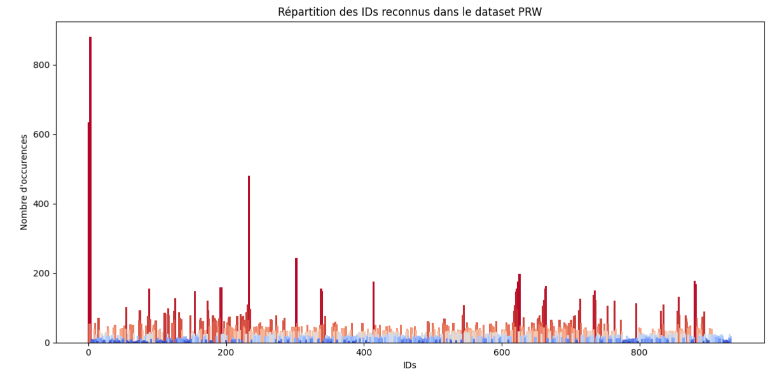
Pedestrian description :

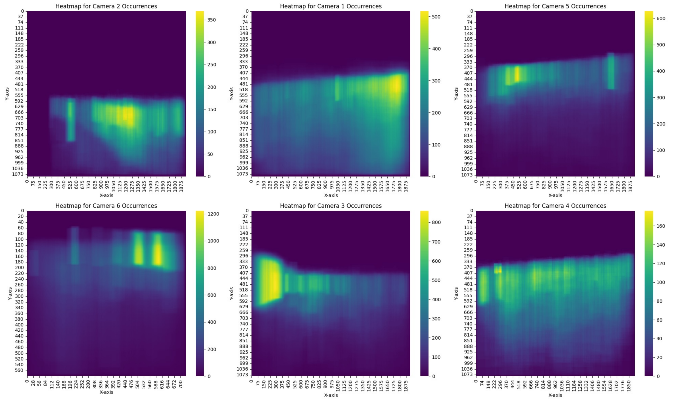
La ré-identification s'effectue en plusieurs étapes. Supposé que l'on possède un dataset d'images ou de vidéos à analyser, la première étape est d'identifier et de reconnaître tous les individus présents sur nos images données en entrée et de les isoler. Pour cela, de nombreuses méthodes de détection existent afin d'isoler les cibles à analyser et identifier, voici une description rapide des plus connues et utilisées :

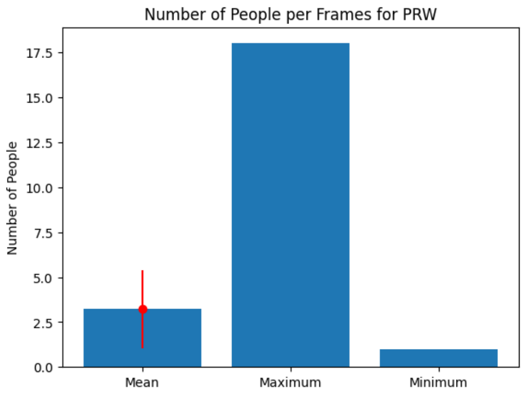
Weighted Color Histogram (WCH) : On isole d'abord le passant dans une image réduite, puis on essaie au plus possible de retirer l'arrière-plan pour ne conserver que le piéton en utilisant des axes de symétrie sur les parties de son corps et en donnant plus de poids aux pixels proches de ces axes. Cela permet de déterminer ce qui semble être en arrière-plan et que l'on peut retirer de l'analyse grâce à des probabilités. Enfin, on range les pixels par couleur afin de tirer un histogramme des couleurs dominantes les plus affluentes sur un cas précis.

Results obtained : Step 2

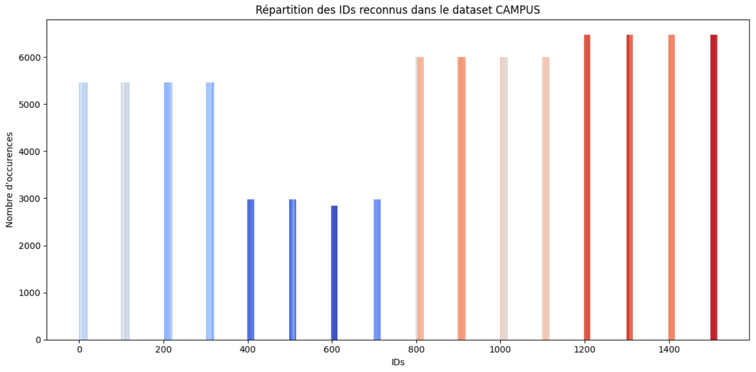
PRW

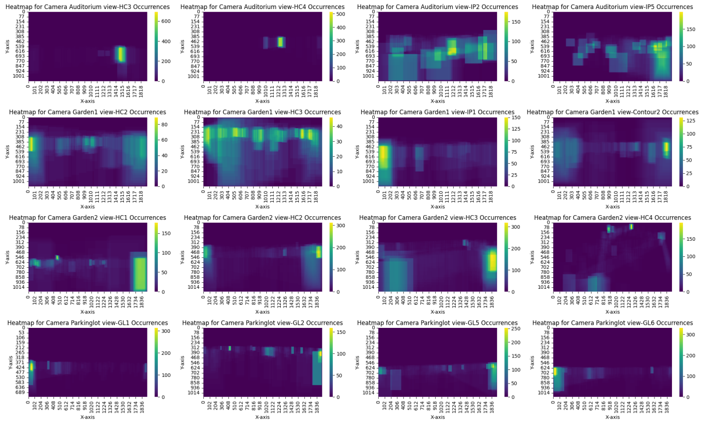


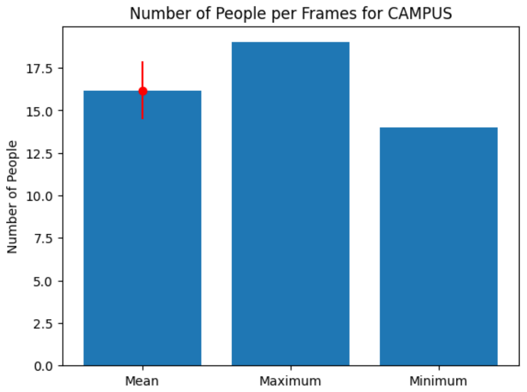




CAMPUS



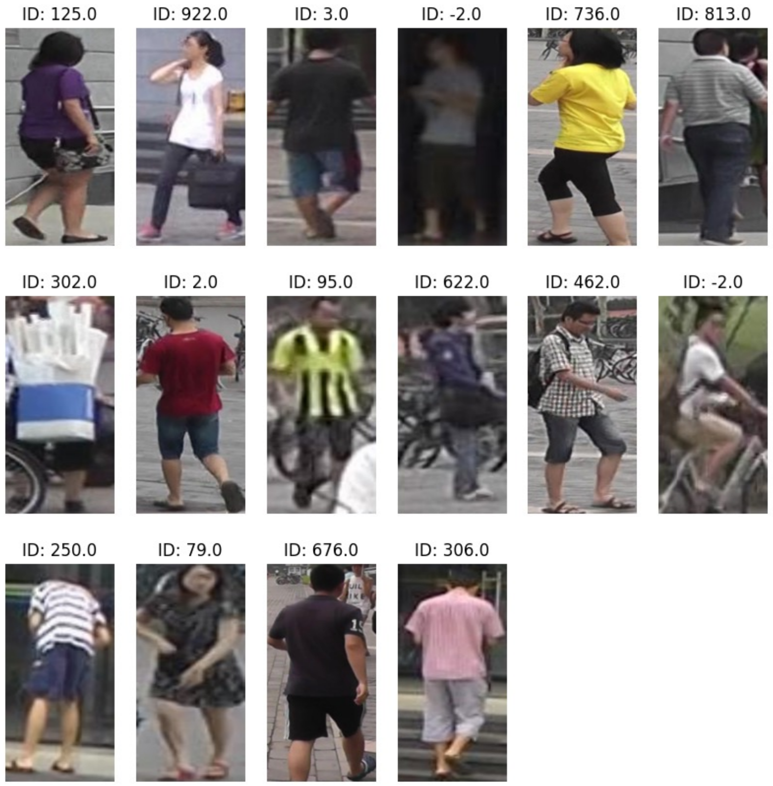




Results obtained : Step 2

```python
class CustomDataGen2(tf.keras.utils.Sequence):
    def __init__(self, df,
                 batch_size=16,
                 input_size=(64, 128, 3),
                 shuffle=True,
                 data_augmentation=False):
        self.shuffle = shuffle
        self.df = df
        self.batch_size = batch_size
        self.input_size = input_size
        self.data_augmentation = data_augmentation

        self.imageGenerator = ImageDataGenerator(
            horizontal_flip=True,
            fill_mode='nearest',
            zoom_range=(1, 1.1),
            width_shift_range=0.1,
            height_shift_range=0.1,
            rotation_range=10
        )

        self.image_paths = df['img_path'].values
        self.labels = df['id'].values.astype(np.float32)
        self.x1_values = df['x1'].values.astype(np.int32)
        self.y1_values = df['y1'].values.astype(np.int32)
        self.x2_values = df['x2'].values.astype(np.int32)
        self.y2_values = df['y2'].values.astype(np.int32)

        self.n = len(self.df)
        self.indexes = np.arange(self.n)

        self.images = []
        for i in range(len(self.image_paths)):
          x1 = self.x1_values[i]
          y1 = self.y1_values[i]
          x2 = self.x2_values[i]
          y2 = self.y2_values[i]
```

```python
          image = Image.open(self.image_paths[i])
          image = image.crop((x1, y1, x1 + x2, y1 + y2))
          image = image.resize(self.input_size[:2])
          image = np.array(image)
          self.images.append(image)

        self.images = np.array(self.images)

        if shuffle:
            np.random.shuffle(self.indexes)

    def __getitem__(self, index):
        images = []

        for i in range(self.batch_size):
            image = self.images[index * self.batch_size + i]

            if self.data_augmentation:
                image = self.imageGenerator.random_transform(image)

            images.append(image)

        indexes = self.indexes[index * self.batch_size:(index + 1) * self.batch_size]
        labels = to_categorical(self.labels[indexes], num_classes=df_prw['id'].nunique())

        return np.array(images), labels

    def __len__(self):
        return len(self.indexes) // self.batch_size

    def on_epoch_end(self):
        if self.shuffle:
            np.random.shuffle(self.indexes)
```



Results obtained : Step 3

Build the model based on Resnet50

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-appli
94765736/94765736 [==============================] - 0s 0us/step
Model: "custom_resnet50"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resnet50 (Functional)       (None, 4, 2, 2048)        23587712

 flatten (Flatten)           multiple                  0

 dense (Dense)               multiple                  8192500

 batch_normalization (Batch  multiple                  2000
 Normalization)

 dropout (Dropout)           multiple                  0

 dense_1 (Dense)             multiple                  467433

=================================================================
Total params: 32249645 (123.02 MB)
Trainable params: 32195525 (122.82 MB)
Non-trainable params: 54120 (211.41 KB)
_________________________________________________________________
```

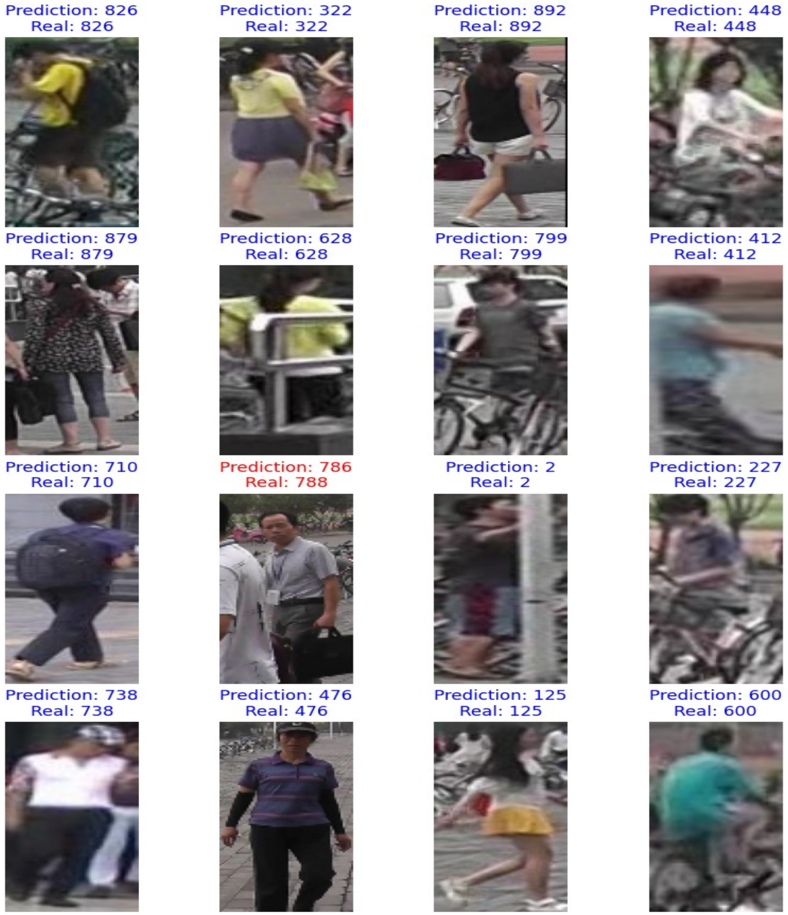
Parameters to fit the model

PARAMETER

```python
EPOCHS = 50
BATCH_SIZE = 32

model.compile(
    tf.keras.optimizers.Adam(learning_rate=1e-3),
    loss = ['categorical_crossentropy'],
    metrics = ['accuracy']
)

plateau = tf.keras.callbacks.ReduceLROnPlateau(
      monitor='val_loss', factor=0.3, patience=2, verbose = 1
)

# Early stopping (stops training when validation doesn't improve for {patience} epochs)
es = tf.keras.callbacks.EarlyStopping(
    monitor='val_loss',
    mode='min',
    verbose=1,
    patience=5
)
```

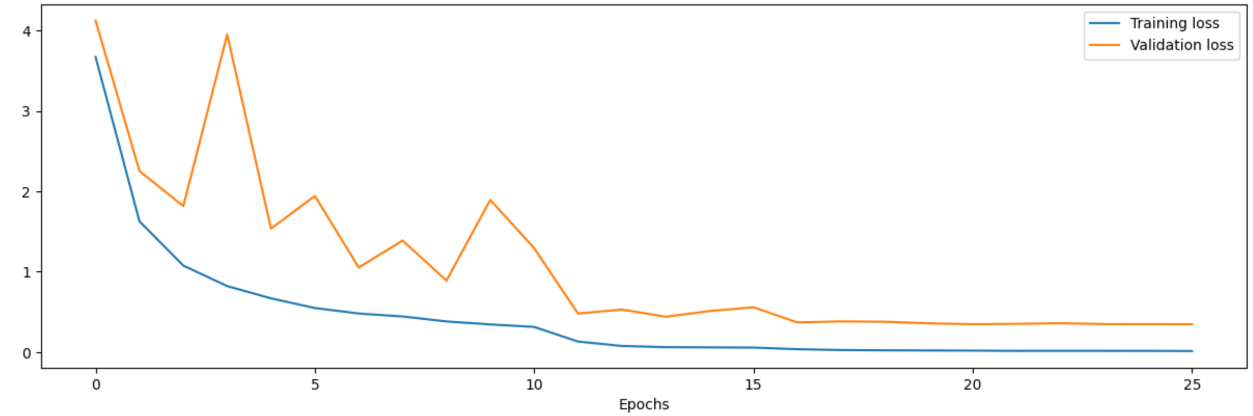
```
Epoch 23/50
848/848 [==============================] - ETA: 0s - loss: 0.0154 - accuracy: 0.9960
Epoch 23: ReduceLROnPlateau reducing learning rate to 8.100000013655517e-06.
848/848 [==============================] - 98s 115ms/step - loss: 0.0154 - accuracy: 0.9960 - val_loss: 0.3584 - val_accuracy: 0.9295 - lr: 2.7000e-05
Epoch 24/50
848/848 [==============================] - 98s 115ms/step - loss: 0.0147 - accuracy: 0.9963 - val_loss: 0.3481 - val_accuracy: 0.9324 - lr: 8.1000e-06
Epoch 25/50
848/848 [==============================] - ETA: 0s - loss: 0.0145 - accuracy: 0.9966
Epoch 25: ReduceLROnPlateau reducing learning rate to 2.429999949526973e-06.
848/848 [==============================] - 97s 115ms/step - loss: 0.0145 - accuracy: 0.9966 - val_loss: 0.3469 - val_accuracy: 0.9330 - lr: 8.1000e-06
Epoch 26/50
848/848 [==============================] - 98s 116ms/step - loss: 0.0127 - accuracy: 0.9970 - val_loss: 0.3465 - val_accuracy: 0.9315 - lr: 2.4300e-06
Epoch 26: early stopping
```

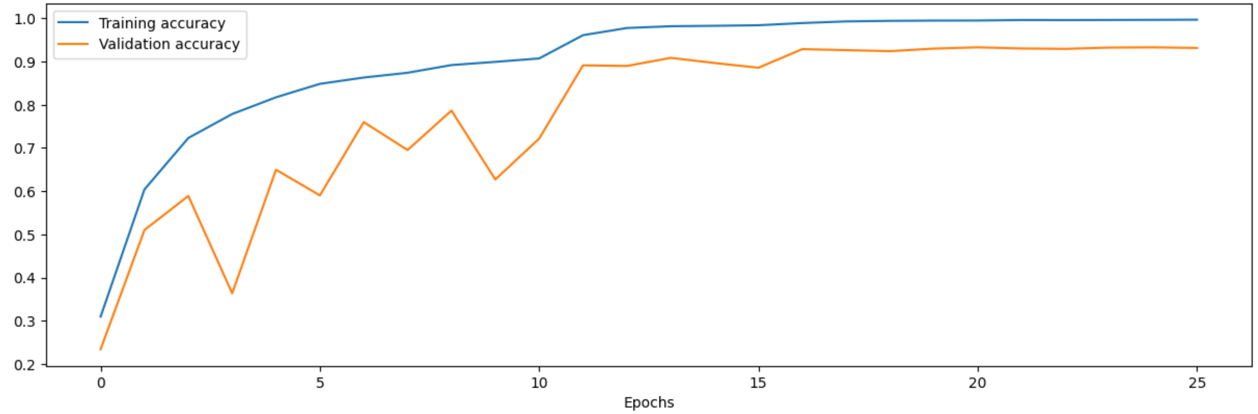
Results obtained : Step 3



Key indicators

```
----- Classification Report -----
Misclassified samples: 256
Accuracy: 0.92
Precision: 0.88
Recall: 0.88
F1_score: 0.87
Kappa: 0.92
```





Next steps

- Implement a tracker focused on drawing bounding boxes on people
- Applying it our model's weights to perform the actual Re-ID
- Find new videos to apply our model to

Organization of the team and distribution of tasks

MBTI Team Radar

I S T/F P

Know how to work in a team

Be effective in communication

Ensure mutual respect for each other's functioning

Myers-Briggs Type Indicator® (MBTI®)

Communication & Tasks

Internal communication and exchange of administrative information, tasks to be carried out.

WhatsApp

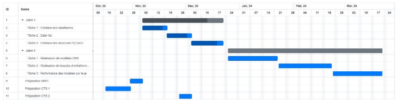
Set of files to work on and holding meetings

Microsoft Teams

Working on a collaborative file

Google Colaboratory

Work planning : Gantt chart



Perspectives

- Lead a project over two semesters with a team of 6 people responding to a partner's need
- Discover the field of Re-Id
- Better understand our deep learning course
- Understanding the current challenges of re-id

Artificially generated image using DALLE

Resources

- Xu, Y., Liu, X., Liu, Y., & Zhu, S. C. (2016). Multi-view people tracking via hierarchical trajectory composition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4256-4265).[En ligne].[Consulté le 03/12/2023]
  Available here: https://bitbucket.org/merayxu/multiview-object-tracking-dataset/src/master/
- Xu, Y., Liu, X., Qin, L., & Zhu, S. C. (2017, February). Cross-view people tracking by scene-centered spatio-temporal parsing. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 31, No. 1).[En ligne].[Consulté le 03/12/2023]
  Available here: https://bitbucket.org/merayxu/multiview-object-tracking-dataset/src/master/
- Zheng, L., Zhang, H., Sun, S., Chandraker, M., Yang, Y., & Tian, Q. (2017). Person re-identification in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1367-1376).[En ligne].[Consulté le 03/12/2023]
  Available here: http://zheng-lab.cecs.anu.edu.au/Project/project_prw.html