

PROJET DE MACHINE LEARNING AVANCÉ : CLASSIFICATION D'IMAGES

- 1) Le type d'architecture utilisé, pour résoudre le problème de la classification d'images, est l'architecture CNN (*Convolutional neural networks*). Puisqu'il est très performant pour le traitement d'images. Le réseau de neurone convolutionnel est constitué de trois couches suivant :
 - La couche convolutionnel utilisant des filtres pour réaliser des opérations de convolutions. Cela va permettre aux réseaux de neurones de détecter les caractéristiques sur les images grâce à l'analyse des images par zones ;
 - La couche de pooling permettant de conserver les informations les plus importantes dans le but de réduire la dimension;
 - La couche fully-connected étant la dernière couche reliant les neurones de la couche précédente afin de classifier l'image donnée en entrée du réseau.
- 2) Notre architecture de CNN contient 3 couches de convolution car 2 couches n'étaient pas suffisantes pour avoir de bonnes prédictions et pour la détection des features sur les images. De plus, l'architecture de CNN avec 4 couches prend trop de temps de calcul et il consomme beaucoup de ressources.
- 3) Pour éviter l'over-fitting, nous avons augmenté notre dataset car ils y avaient 2 fois plus de valeurs pour 1 comparé à -1, nous avons donc augmenté les valeurs de -1 en faisant de la data augmentation et nous avons vérifié le résultat de chaque entraînement en fonction des epochs et des paramètres (en limitant le nombre d'epoch car trop d'epoch peut entraîner de l'over-fitting).
- 4) L'asymétrie de la base de données est un problème car notre modèle aura plus de facilité à reconnaître un cas comparé à un autre, cela peut entraîner de l'overfitting. Pour éviter d'avoir une base de données asymétrique, nous avons augmenté les données des valeurs de -1 en faisant des flipLR() sur plus de la moitié des images dont la valeur est -1 (-1 possède 2 fois moins de données que 1).

Le choix de la fonction flipLR() a été faite pour une raison très importante. puisque nous ne connaissons pas en amont le critère de classification. Si le label -1 représentait la position de la personne sur l'image, nous ne pouvions pas faire de rotation ou décaler la position de la personne sur l'image. Cependant, lors de nos tests, nous avons eu des résultats moins intéressants en utilisant flipLR. De ce fait, nous avons décidé de ne pas l'intégrer lors de l'entraînement.

```

# data augmentation + df to new df
def rotate(self):
    list = self.label_list[self.label_list['type'] == 0] # list of all value == 0
    list = list.head(n=int(list.shape[0]/2))
    it = 0 # iterator

    # flip image + add to new dataframe
    for index, rows in list.iterrows():
        img = cv2.imread(self.directory + rows['id'])
        flipLR = np.fliplr(img) # change image from left to right

        self.df.loc[it] = [img, rows['type']] # add img to df
        it += 1

    # add former df to new df
    self.new_df(it)

# old df to add new df with loaded img
def new_df(self, it):
    # add former df to new df
    for index, rows in self.label_list.iterrows():
        img = cv2.imread(self.directory + rows['id'])
        self.df.loc[it] = [img, rows['type']] # add value to new df
        it += 1

```

- 5) Les labels bruités ne permettent pas au modèle de générer une bonne prédiction. Autrement dit, il ne classe pas correctement ces deux classes. Pour éviter ce problème, nous avons bien entraîné le modèle afin qu'il puisse classer les images de manière cohérente.
- 6) Nous avons utilisé la fonction `torch.max()` de la librairie `torch`. A partir de la sortie du réseau de neurone si la valeur est proche de 1, nous lui attribuons la valeur de 1 et si elle est proche de 0, nous lui attribuons 0 (cela correspond au -1 dans notre cas).

```

# Get predictions from the maximum value
_, predicted = torch.max(outputs.data, 1)

```

- 7) Si on avait eu un mois supplémentaire, nous aurions pu supprimer les images bruitées afin d'aider le modèle à mieux se généraliser. Nous aurions pu tester d'autres paramètres car par défaut de temps et de ressources, certaines options n'ont pas été utilisées. Avec le délai supplémentaire, nous aurions pu tester/trouver d'autre méthode de data augmentation.