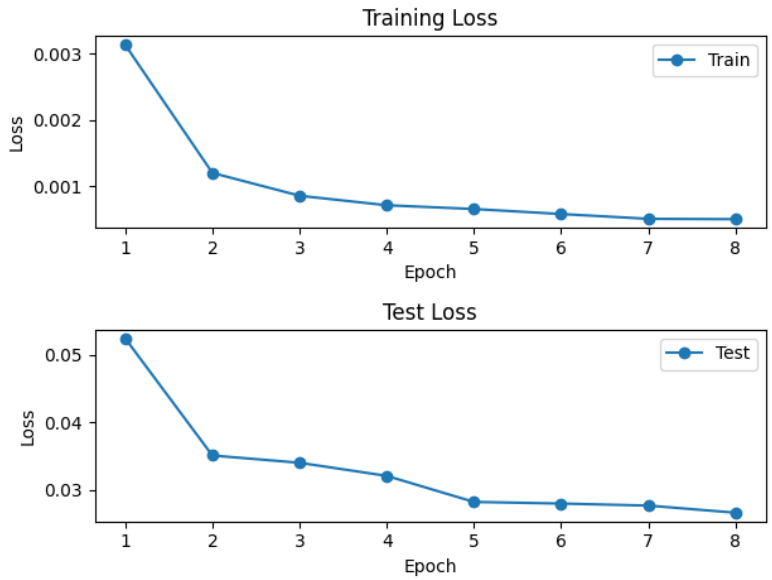
Delestre Gwendoline
M2 MIASHS IC

TP noté Perception Multimodales

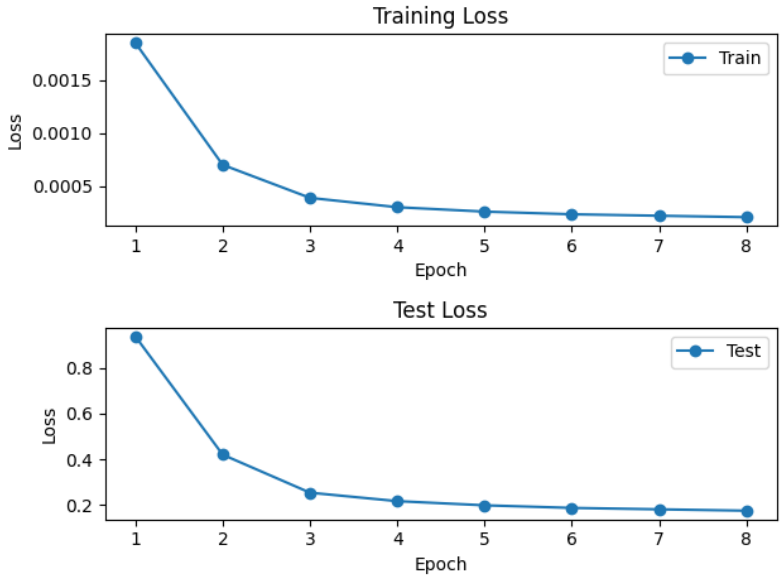
Exercice 1 : MNIST

1) Le code main_1.py a été lancé sur 8 époques. Voilà les résultats obtenus :



On voit sur le graphe Training Loss que la moyenne des loss de la première époque est bien plus élevée que pour les autres époques, cette moyenne diminue au fur et à mesure que les époques passent. On observe la même tendance pour Test Loss avec une diminution moins forte entre les époques 1 et 2. L'échelle des deux graphiques est différente, bien plus petite pour le jeu d'entrainement.

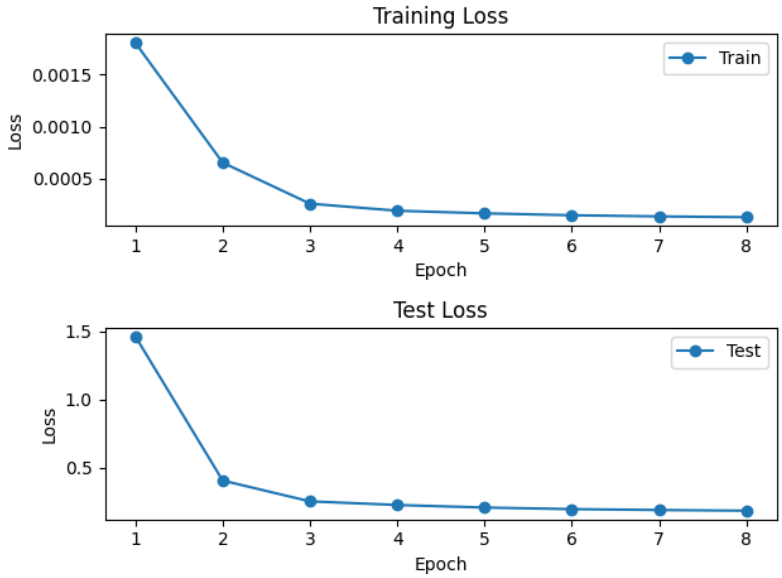
2) Le code main_2.py a été lancé sur 8 époques. Les corpus train et test ont été inversés. Voilà le résultat :



On observe les mêmes tendances que pour le graphique 1 pour les deux jeux de données mais la moyenne de loss de la première époque du jeu Test est bien plus importante que lorsqu'on avait beaucoup de données pour l'entrainement. En revanche, pour le jeu Train, la moyenne des losse est bien inférieure dès la première époque.

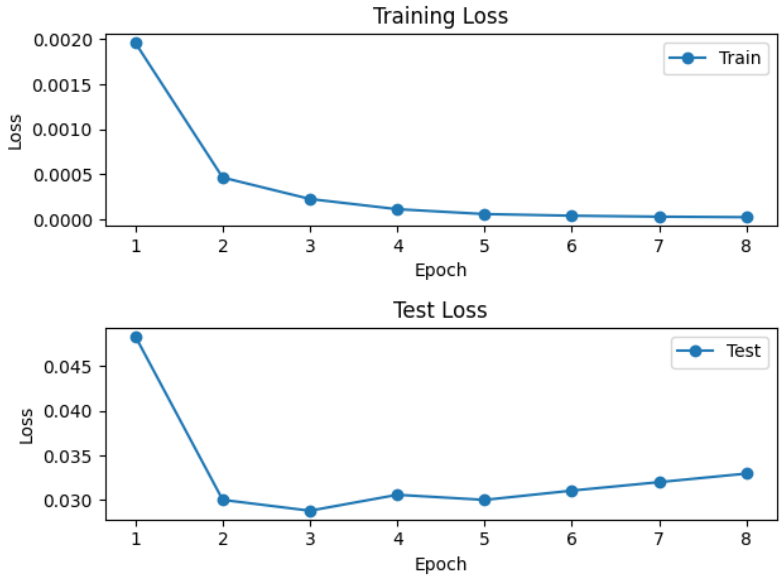
Delestre Gwendoline
M2 MIASHS IC

3) On lance le code main_3.py sur 8 époques. On garde l'inversion des jeux de données et on supprime les fonctions de dropout. Ces fonctions empêchent le sur ajustement des données du jeu Train en abandonnant des neurones (dans un réseau de neurones évidemment) avec une probabilité *p* [1]. Voilà le résultat :



Ces résultats sont très similaires avec celui du script précédent mais la moyenne de la loss diminue beaucoup plus vite, dans le sens où la différence entre la première époque et la deuxième est plus importante. La moyenne des loss du jeu de Test est encore plus élevée que lorsqu'on avait les fonctions de dropout alors qu'elle est un peu près équivalente pour le jeu de Train.

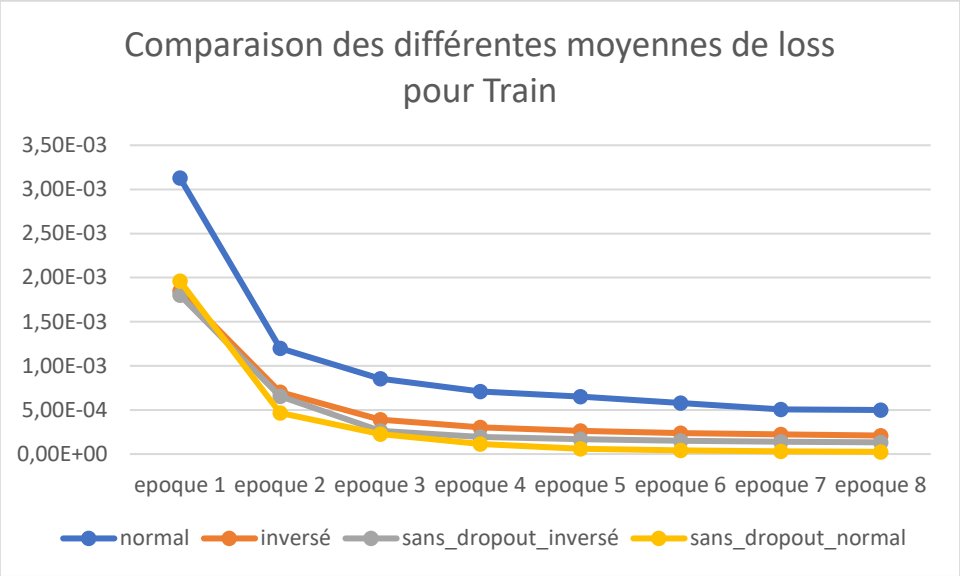
4) On lance le script main_4.py avec toujours 8 époques. On remet les corpus comme pour la question 1 mais on ne remet pas les fonctions dropout. Voilà le résultat :



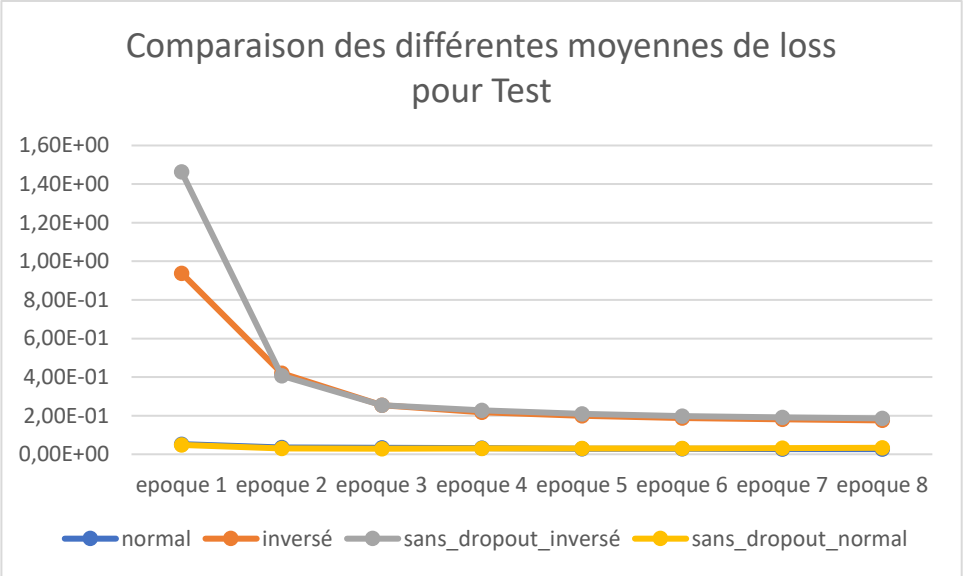
On observe que la moyenne de loss de l'époque 1 est bien inférieur à ce qu'on a déjà vu pour le jeu de Train. De même pour le jeu de Test. En revanche, on observe un sur-apprentissage pour le jeu de Test, la moyenne de loss étant en augmentation sur les 4 dernières époques.

Delestre Gwendoline
M2 MIASHS IC

5) On synthétise nos résultats sur deux graphiques, un pour les données d'apprentissage, un celles de test.



On observe ici que la moyenne de loss du premier script est plus élevée que pour les autres et se rapproche le moins de 0. Au contraire, lorsque l'on a enlevé les fonctions de dropout, cela a permis de fortement diminuer la moyenne de loss.

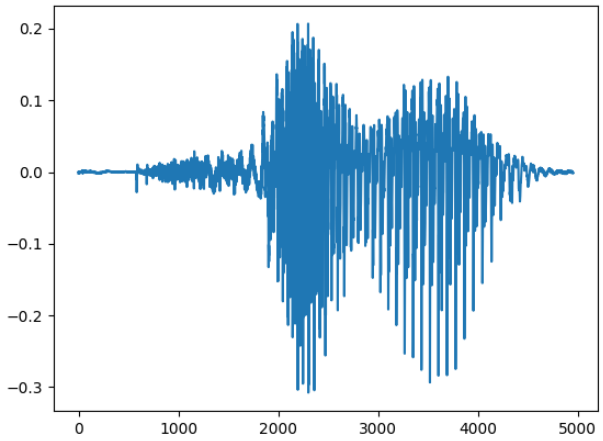


Sur les données de Test, on voit qu'il est bien plus mauvais lorsqu'on a inversé les corpus sans surprise puisque les données de Train sont bien moins nombreuses. En revanche pour les corpus normaux, avec les dropout, on observe une moyenne faible de loss et une diminution au fur et à mesure des époques. Lorsqu'on enlève les dropout, on observe un léger sur-apprentissage. Compte tenu du grand nombre de données et de l'absence de fonctions dropout cette fois-ci, le système a « trop » appris les données de Train et ne peut pas suffisamment généraliser.

Delestre Gwendoline
M2 MIASHS IC

Exercice 2

0_lucas_49.wav :



D'après le cours, le temps est représenté sur l'axe des abscisses, et sur l'axe des ordonnés se trouve la fréquence en hertz. On a donc l'amplitude du son en hertz en fonction du temps.

Ici, on a une représentation de MelSpectrogram, une version spécifique du spectrogramme. Les fréquences sont converties en fréquence de Mel. Les basses fréquences sont plus espacées, tandis que les hautes fréquences sont plus rapprochées, de cette façon, les fréquences sont représentées de manière à mieux coller à la façon dont nous percevons le son.

Le CSV test comporte 750 échantillons, celui de train comporte le reste des échantillons. Il y a bien plus de données dans le jeu de train ce qui permet de bien généraliser par la suite puisque le modèle peut apprendre plein d'exemple différent et d'obtenir de meilleurs résultats lorsque nous appliquons le modèle au jeu de test.

Les blocs 11 à 15 :

11 : Création de la classe CNN.

12 : Permet d'utiliser le modèle sur le GPU s'il existe.

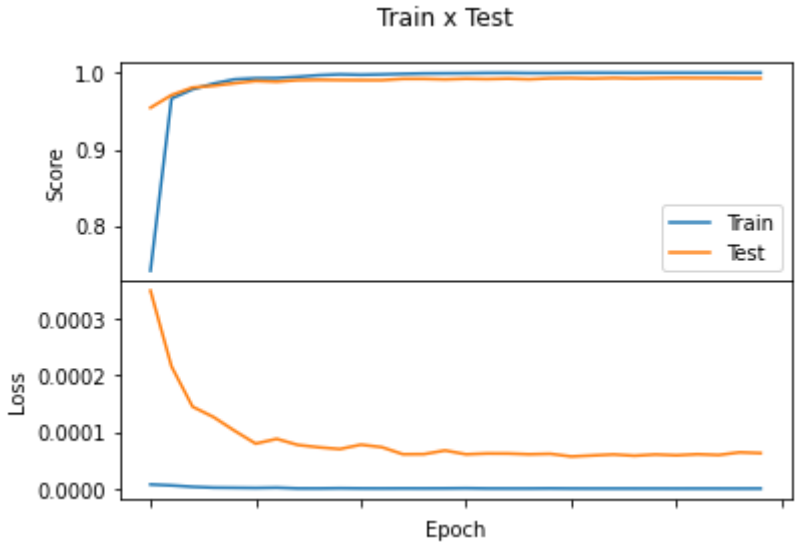
13 : Utilisation de CrossEntropyLoss utilisée pour la classification parmi plusieurs classes possible lorsque les classes égales [2]. Puis utilisation de optim.SGD qui une descente de gradient stochastique [3].

14 : Définition des fonctions pour l'entrainement du modèle et l'application du modèle sur le jeu de test.

15 : Entrainement du modèle sur les données de train et de test sur 30 époques.

Ils décrivent un modèle CNN avec deux couches de convolution puis une couche de ReLU après chaque convolution et une couche linéaire.

Delestre Gwendoline
M2 MIASHS IC



Sur ce graphique, on voit que pour les losses de Train sont très faibles quel que soit l'époque contrairement à ceux de Test mais le score est tout de même plus élevé pour le Test que pour le Train à l'époque 1. Malgré une différence de losses, le score reste similaire pour les deux jeux après la première époque.

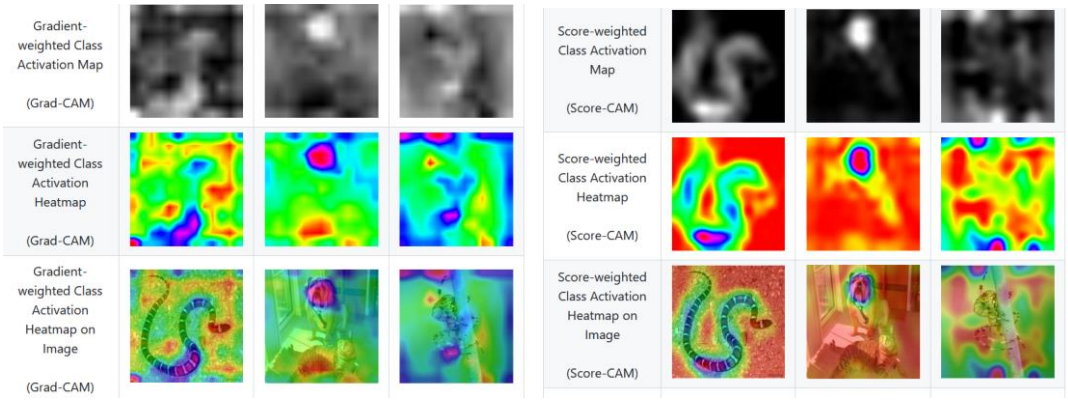
Pour réaliser un système multimodal de reconnaissance de caractères où l'on écrirait et énoncerait le chiffre en même temps A partir des corpus Mnist et audioMnist on pourrait faire une fusion de données traitant les images et, séparément l'audio avec des modèles différents (LSTM et FC par exemples) et après concaténer les features.

Au lieu d'utiliser un MelSpectrogramme, on pourrait utiliser la lecture sur les lèvres à partir d'une vidéo, reconnaître les sons en fonction du mouvement de la bouche. Ensuite nous pourrons fusionner les données comme expliqué plus haut.

Exercice 3 :

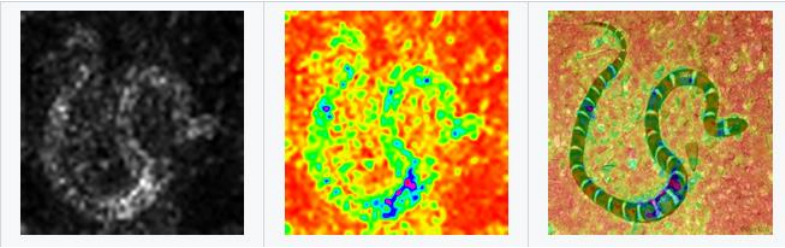
Q1)

Gradient Visualization : représentation graphique des gradients



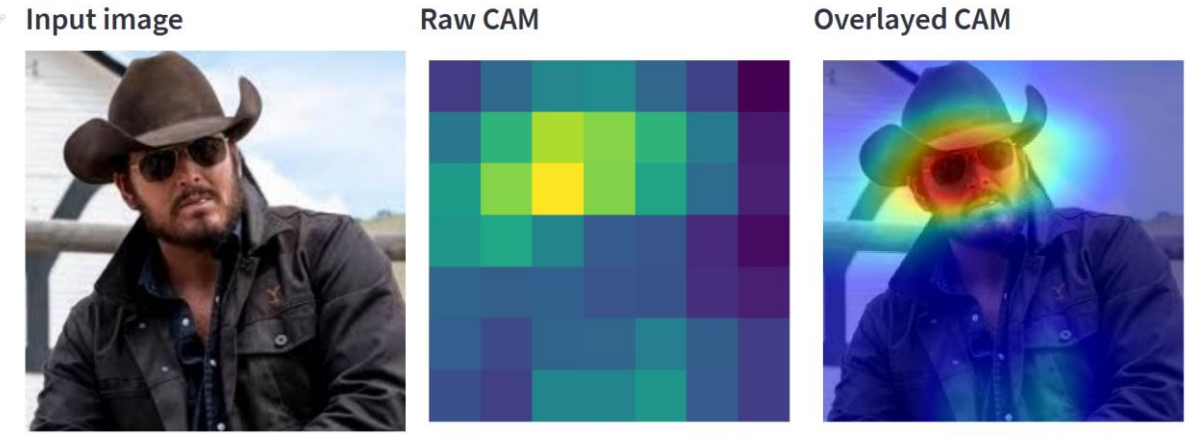
Delestre Gwendoline
M2 MIASHS IC

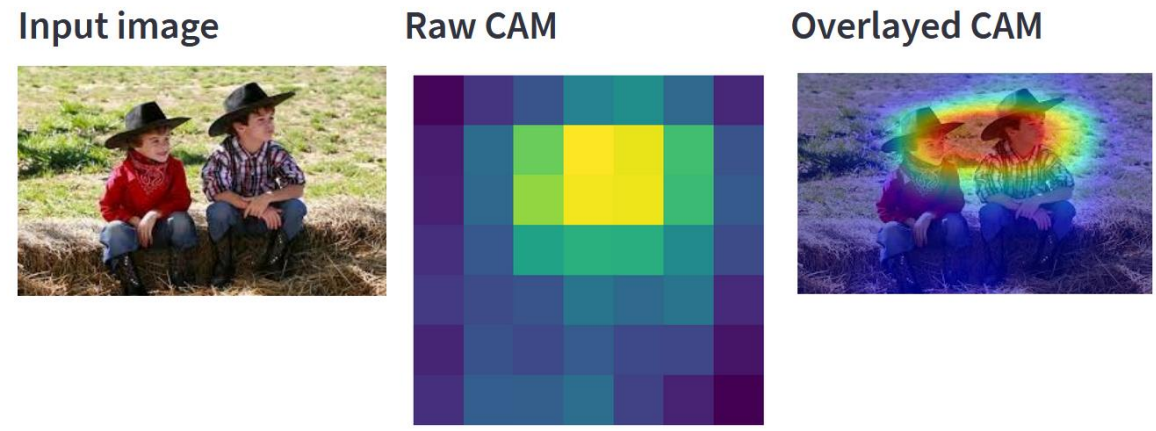
Hierarchical Gradient Visualization : gradient visualization qui prend en compte la hiérarchie

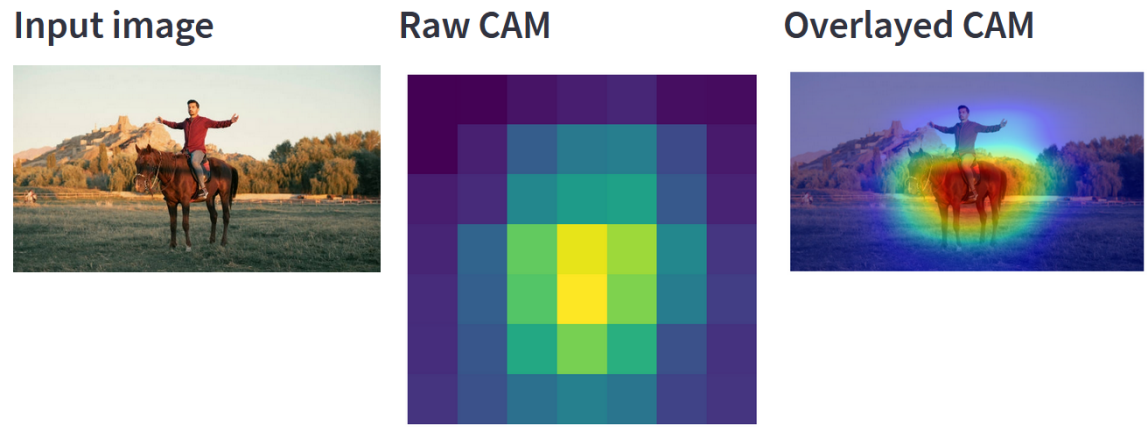


Q2) et Q3)

Paramètres choisis : 516 - Cowboy hat, resnet_18, CAM





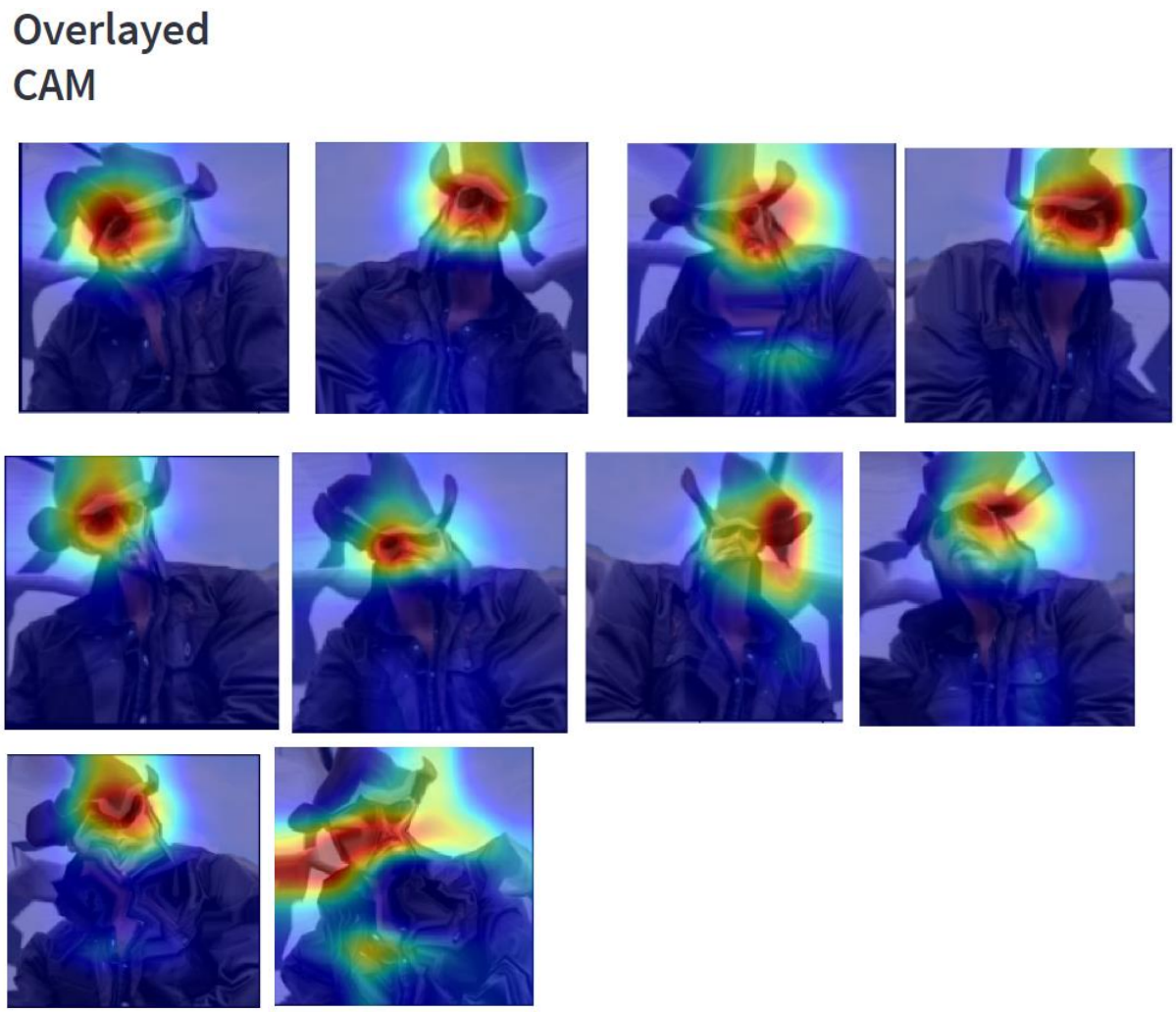


Delestre Gwendoline
M2 MIASHS IC

Lorsque qu'un ou plusieurs chapeaux de cowboy se trouvent sur l'image, le modèle n'est pas vraiment efficace pusiqu'il ne donne pas le chapeau dans la première image mais les lunettes du cowboy qui sont juste en dessous et dans la deuxième bien qu'intégrant une petite partie des chapeaux des enfants, il indique surtout leur visage et l'espace entre les deux enfants, il ne distingue pas deux objets. Pour la troisème image, sans chapeau de cowboy, il détecte le cheval, rien a voir avec un chapeau.

Q4)

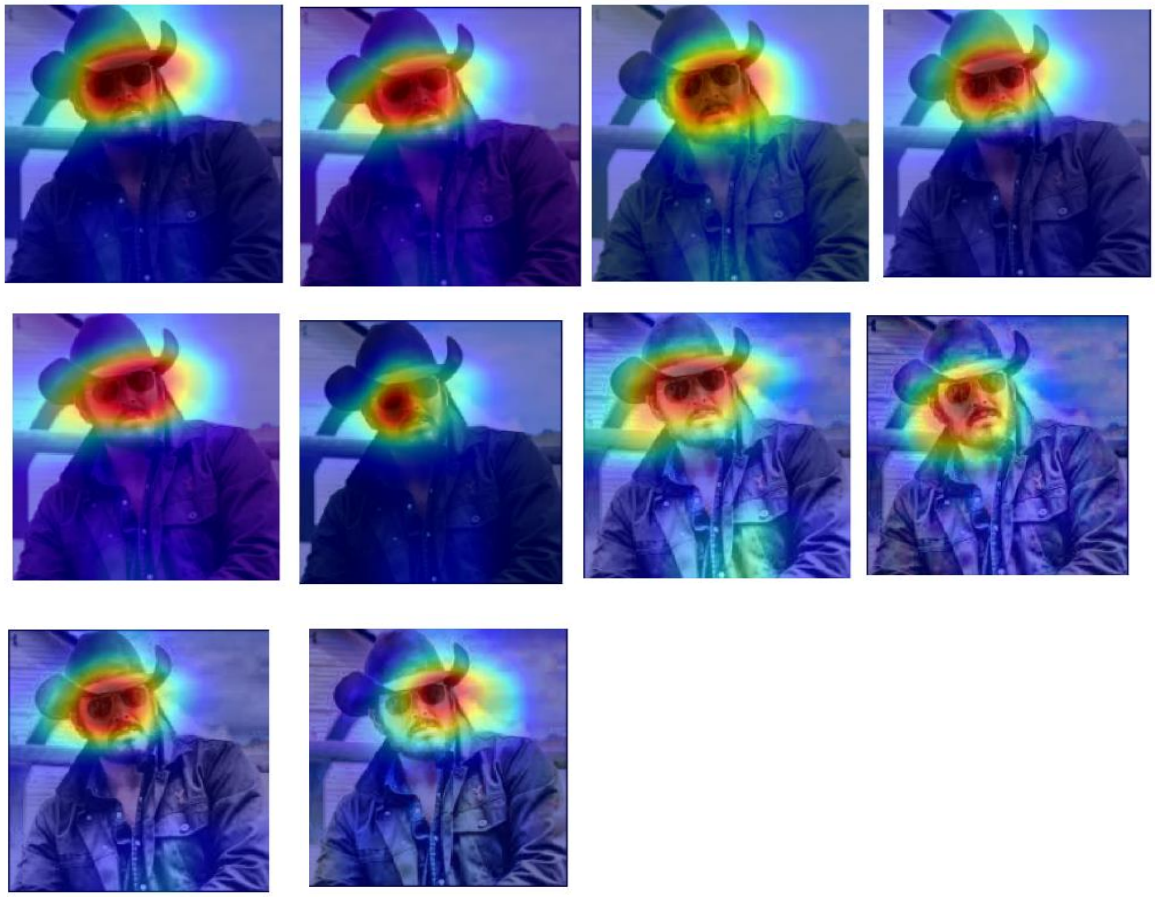
Distorsion :



On voit après distorsion de l'image que le modèle à toujours du mal à trouver le chapeau, il donne le plus souvent un morceau des lunettes ou une partie plus ou moins étendue du visage, sauf pour la dernière image où il ne comprend rien. Il trouve même deux objets.

Delestre Gwendoline
M2 MIASHS IC

Contraste :



Lorsqu'on joue avec le contraste, le modèle n'est pas bon mais il ne donne quand même 8 fois sur 10 environ la même chose que pour l'image de base, deux fois il a donné seulement un morceau des lunettes.

Q5)

Pour conclure, on peut dire que les réseaux de neurones convolutionnels sont sensibles aux variations d'images notamment à la distorsion puisque le jeu de donnés d'entrainement ne doit pas contenir d'image déformer de la sorte. En revanche, le contraste de l'image ne semble pas déranger plus que ça le réseau, les images du jeu d'entrainement devaient être de contraste assez différent.