

I. CNN & RNN pour la reconnaissance de mouvements/actions

1.1. Partie théorique :

- Expliquer la différence entre un réseau de neurones convolutionnel (CNN) et un réseau de neurones récurrent (RNN))
- Expliquer le problème de « Vanishing Gradient » rencontré avec les RNNs. Quelles solutions existent ?

1.2. Partie Pratique :

- Développer un réseau de neurones profond utilisant les CNNs et RNNs en vue de détecter les mouvements ou actions dans une séquence vidéo. Vous pouvez réaliser les expérimentations à l'aide de bases de données publiques.

1.3. Liens intéressants :

- Méthode [TSM: Temporal Shift Module for Efficient Video Understanding](#)
- Code : <https://github.com/open-mmlab/mmdetection>
- Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*.

II. Réseaux de neurones génératifs GAN pour la prédiction de mouvements

2.1. Partie théorique :

- Expliquer le principe des réseaux de neurones de type « Auto encoder »
- Expliquer le principe des réseaux de neurones génératifs « GAN »
- Quelle est la différence entre réseaux « Auto encoder » et « GAN » ?

2.2. Partie Pratique :

- Développer un réseau de neurones de type « GAN » pour prédire les mouvements et actions (frames futurs) dans une séquence vidéo dans le domaine de vidéosurveillance. Par exemple, si une personne se dirige vers une zone interdite, le réseau permettra de prédire que la personne se retrouvera en zone interdite dans quelques instants.

2.3. Liens intéressants :

- Paper 1 "[Predicting Future Frames using Retrospective Cycle GAN](#)"
- Paper 2 : "[BiHMP-GAN: Bidirectional 3D Human Motion Prediction GAN](#)"
- <https://github.com/amoghadishesha/GAN-motion-Prediction>
- <https://paperswithcode.com/task/human-motion-prediction>

III. Auto-encoders pour la segmentation d'images médicales « Covid-19 »

3.1. Partie théorique :

- Expliquer le principe des réseaux de neurones profonds utilisés pour la segmentation d'images ;
- Illustrer quelques architectures neuronales utilisées pour la segmentation d'images médicales.

3.2. Partie Pratique :

- Développer un réseau de neurones profond permettant de segmenter les poumons et lésions responsables de la pathologie Covid-19 en utilisant des images de scan (CT-scans).
- Vous pouvez travailler avec des bases de données publiques.

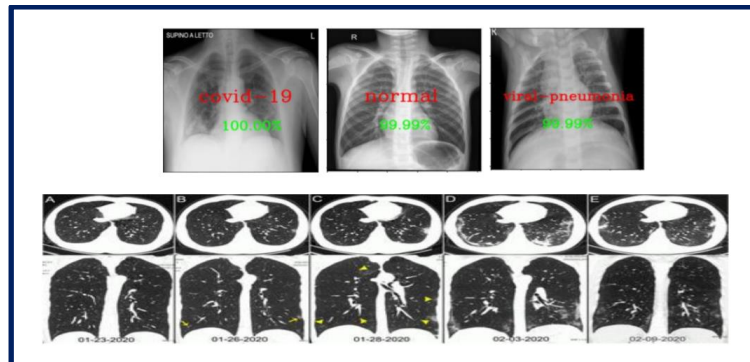


Figure 1: exemple d'images X-ray et CT-scans : covid-19, normal et viral-pneumonia

IV. Liens intéressants :

- Exemple de base de données : <https://github.com/UCSD-AI4H/COVID-CT>
- <https://paperswithcode.com/task/covid-19-image-segmentation>

IV. XAI pour la classification et reconnaissance d'actions

4.1. Partie théorique :

- Dans la littérature, les méthodes de Deep Learning sont appelées par le terme “**black-box model**”. Quelle en est la signification ?
- Proposez une méthode d'explicabilité de l'intelligence artificielle permettant d'expliquer les décisions d'un réseau Deep Learning entraîné. Expliquer son fonctionnement.
- Un modèle Deep Learning entraîné à une précision de 100% sur un ensemble de test. Cela est-il suffisant pour justifier son utilisation directe dans un environnement réel ? Proposer quelques idées (liées ou non à l'explicabilité) et justifiez-les.

4.2. Partie Pratique :

- Développer une méthode d'explication et d'interprétation de modèles Deep Learning appliqués à une application de classification de vidéos et de reconnaissance d'actions. La particularité de ce type d'application est représentée par son besoin de combiner l'information spatiale (image)

et temporelle présentes dans une séquence vidéo ce qui requiert le développement de modèles combinant différents types de réseaux de neurones profonds (MLP, CNN, RNN, LSTM, etc.). L'objectif de ce mini-projet d'interpréter et expliquer ces modèles combinant différents DNNs afin de :

- Calculer et quantifier la contribution chaque frame (image) de la vidéo dans le résultat final du modèle : détecter les frames pertinents.
- Parmi les frames pertinents, calculer et quantifier la contribution des pixels (ou régions) dans le résultat final.

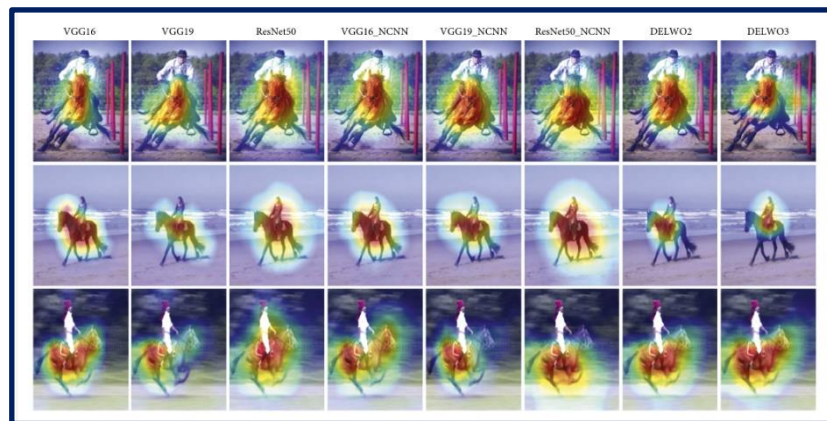


Figure 2: exemples de reconnaissances d'action avec carte d'attention [Paper1]

4.3. Liens intéressants :

- [Paper 1](#)
- [Paper 2](#)
- [Paper 3](#)
- [Code](#)

V. Système « Edge IA » pour villes intelligentes

5.1. Partie théorique :

- Présenter une analyse comparative entre les différents types de réseaux de neurones profonds

5.2. Partie pratique :

L'objectif de ce mini-projet est de développer un système IA embarqué sur une ressource Edge ([nvidia Jetson Xavier](#)). Le système s'appuiera sur les techniques et modèles Deep Learning de classification d'images et de localisation d'objets qui peuvent être utilisées dans un système de ville intelligente. Ces modèles seront combinés pour fournir un module « Edge IA » appliqué aux vidéos capturées en temps réel. Vous pouvez intégrer deux ou trois modèles parmi :

- Modèle de reconnaissance faciale

- Modèle de détection de feux de forêts : pour détecter les feux ou fumées dans des forêts proches de la ville intelligente en utilisant les images capturées via la caméra. Une base de données et une solution sera fournie
- Modèle de détection d'objets suspects : pour détecter la présence d'un ou plusieurs objets suspects ou non autorisés (ex. bâtons, sacs isolés, armes, etc.) dans une scène filmée par la caméra.

Le challenge sera de porter les réseaux de neurones sur la ressource Edge et fournir un calcul en temps réel. Le choix d'architectures neuronales devra prendre en compte la taille des modèles générés ainsi que leurs temps d'inférence.

Note : le matériel nécessaire (carte Jetson Xavier + accessoires) ainsi que deux modèles (classification d'images de feu et localisation d'objets personnels) seront fournis.

5.3. Liens intéressants :

- Modèle de reconnaissance faciale : <https://github.com/davidsandberg/facenet>

VI. Compression et Optimisation de réseaux de neurones profond

6.1. Partie théorique :

- Citer et expliquer les méthodes de compression de réseaux de neurones profonds

6.2. Partie pratique :

- Développer un algorithme de compression de réseaux de neurones profonds (ex. Pruning, quantification, convolution séparable 1D) permettant de réduire sa taille mémoire, accélérer son temps d'inférence en garantissant le maintien de bonnes performances en termes de précision « accuracy » et perte « loss ».

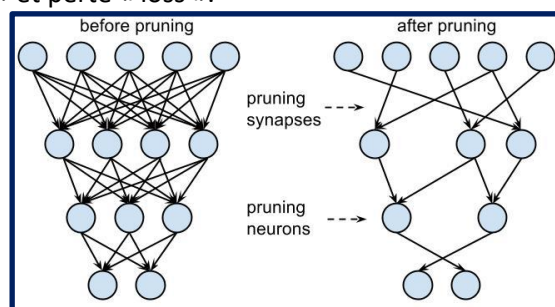


Figure 3: technique de Pruning

6.3. Liens intéressants :

- Code 1: <https://github.com/arturjordao/PruningNeuralNetworks>
- Code 2 : <https://github.com/jiajuns/Neural-Network-Pruning-Keras>
- Paper 1 : <https://arxiv.org/pdf/1710.01878.pdf>
- Paper 2: <https://arxiv.org/pdf/1611.06440.pdf>