Projet PRAT : Rain Nowcasting

Rapport

Lucas Beretti

Encadrant : Dominique Béréziat

13 février 2022

Table des matières

Chapitre 1

Rapport bibliographique

1.1 Introduction

Tout au long de son histoire, l'homme s'est intéressé au domaine de la prévision météorologique du fait de la forte dépendance de ses activités à celui-ci. Aujourd'hui encore, elle joue un rôle central dans de nombreux secteurs météo-sensibles comme l'agriculture, le transport, l'énergie, le tourisme ... Bien que les méthodes d'estimation de la météo se sont largement améliorées au cours des dernières décennies, la fiabilité de ces modèles est un sujet sur lequel les météorologues travaillent encore.

La prévision du taux de précipitation est une donnée cruciale pour l'ensemble des secteurs météo-sensibles. Ces prévisions sont effectuées par des modèles physiques intégrant divers phénomènes liés à la dynamique de l'atmosphère. Bien que ces méthodes apportent de bonnes prévisions de précipitation sur plusieurs jours, elles manquent encore de précision à très court terme, jusqu'à deux heures à l'avance. Cette problématique est plus communément appelée "Rain Nowcasting".

L'émergence des technologies d'apprentissage automatique, et plus précisément d'apprentissage profond, a amené certains chercheurs à s'intéresser à ce problème.

Cet état de l'art a pour but de présenter un panel de méthodes élaborées au cours des dernières années sur des images radar. Ensuite, l'objectif sera d'implémenter et de comparer quelques méthodes sur des données radar de Météo-France.

1.2 Méthodes traditionnelles

Cette partie a pour but de présenter quelques méthodes traditionnelles pour la prévision à court terme du taux de précipitation. De manière abusive, nous qualifierons de méthodes traditionnelles toutes les méthodes établies avant l'émergence de l'intérêt pour les techniques d'apprentissage profond appliquées à la résolution de cette problématique.

1.2.1 Mise en correspondance de blocs d'images

Les plus anciens types d'approches sont celles par mise en correspondance de blocs d'images.

Tracking Radar Echoes by Correlation (TREC) [1] est la première de ce type. Elle consiste à comparer des images de réflectivité des ondes radar d'une même zone acquises à quelques minutes d'intervalle. Chaque image va être découpée en blocs de même taille séparés par une distance donnée. Chaque bloc de la première image acquise va être comparé à l'ensemble des blocs de la seconde. Les coefficients de corrélation entre les motifs de réflectivité dans les blocs sont calculés pour toutes les paires de blocs possibles, et la paire avec le coefficient de corrélation le plus élevé est sélectionnée pour la détermination du vecteur de mouvement qui indique le mouvement du bloc d'origine pendant la période de temps Δt. Une fois le champ de mouvement calculé (c'est à dire que le vecteur de déplacement a été calculé pour l'ensemble des blocs), est alors obtenu un champ de mouvement de réflectivité qui va être utilisé pour estimer le déplacement des blocs, et donc les précipitations, par extrapolation. Ces précipitations sont adaptées à chaque nouvelle acquisition faite par la mise à jour du champ de mouvement de réflectivité.

Cette méthode est très sensible à de nombreux paramètres comme la taille des blocs, la distance séparant le centre de deux blocs, la taille de la grille des données radar, l'intervalle de temps entre deux acquisitions. Elle est efficace pour décrire le mouvement de cellules pluvieuses mais ne permet pas de prendre en compte le mouvement global à plus grande échelle de la structure orageuse. Enfin, le vecteur de mouvement se veut parfois incohérent et bruité. La principale cause est le phénomène de fouillis de sol, lorsque d'autres éléments sont détectés, tels que des oiseaux, des insectes, des objets près du sol et de la poussière. Ces échos de fouillis génèrent des trous dans le champ de mouvement de réflectivité [2].

Des solutions à ces différents problèmes ont été développées. *Continuity of Tracking Radar Echoes by Correlation* (COTREC) vise à corriger le bruit causé par le phénomène de fouillis de sol [2]. Enfin, la méthode *Multi-scale Tracking Radar Echoes by Correlation* (MTREC) intègre de manière plus globale le mouvement de la structure orageuse. En effet, elle consiste en l'application de la méthode TREC avec des blocs de tailles différentes pour décrire les phénomènes locaux et plus globaux [3].

1.2.2 Mise en correspondance de cellules de précipitations

Les méthodes de mise en correspondance de cellules de précipitations sont assez proches, dans leur fonctionnement, de celles de correspondance de blocs d'images. Cependant, au lieu de segmenter l'ensemble de l'image en blocs, un premier algorithme va identifier les cellules et leurs caractéristiques sur l'image. Ensuite, différentes méthodes existent pour la mise en correspondance des cellules identifiées entre deux images.

L'approche *CELLTRACK* va prendre en compte un critère de similarité de forme entre le coeur d'une cellule (la zone de réflectivité maximale au sein d'une cellule pluvieuse) d'une première image et les coeurs des cellules de la seconde. A cela s'ajoute la notion de la distance euclidienne entre les coeurs des cellules. L'algorithme essaye dans un premier

temps d'apparier les clusters de cellules proches car ce sont les plus susceptibles d'être sujet à des séparations et fusions des noyaux de réflectivité. Comme pour les approches par blocs, les prévisions sont effectuées par extrapolation du vecteur de mouvement estimé entre deux acquisitions radar [4].

1.2.3 Méthodes s'appuyant sur le flot optique

Le majeur inconvénient des méthodes précédentes est que le calcul du vecteur de vitesse du déplacement se limite aux blocs ou aux cellules de précipitations identifiées. Ainsi, hormis MTREC qui s'applique à plusieurs échelles, elles semblent plus adaptées à décrire le mouvement de blocs ou de cellules de précipitations individuelles plutôt que le mouvement global.

Les approches par flot optique apportent des solutions à ces problématiques. Elles calculent des vecteurs de déplacement en tout pixel de l'image et utilisent des images à plusieurs échelles pour prendre en compte le comportement météorologique à large et plus fine échelle. [5]

Ces modèles partent de l'hypothèse de la conservation de la réflectivité au fil du temps le long des trajectoires. Cette hypothèse se traduit par une relation à laquelle il n'est pas possible d'obtenir une unique solution de par son nombre d'inconnues. Il est alors nécessaire d'introduire des contraintes supplémentaires sous la forme d'un problème de minimisation d'une fonction de coût, ayant pour but d'assurer la continuité et la régularité du gradient. C'est ce qui est fait dans la méthode *Multi-Scale Optical-Flow by Variational Analysis* (MOVA) [5]. Des fenêtres de différentes tailles sont appliquées pour le calcul du vecteur de déplacement des pixels. L'algorithme est en cascade : une première estimation du champ vectoriel est calculée pour la fenêtre de taille la plus grande correspondant au comportement météorologique dans sa globalité. Ce résultat va être affiné de manière itérative par réduction de la taille de la fenêtre et donc, la prise en compte de phénomènes plus locaux.

La méthode *Real-Time Optical Flow by Variational Methods for Echoes of Radar* (ROVER) (aussi dans [5]) est similaire à MOVA mais ajoute une étape de pré-traitement pour lisser les données et augmente la vitesse des calculs en utilisant une fonction de coût différente.

Les champs de mouvement estimés par ces méthodes sont utilisés pour calculer les prévisions du taux de précipitations par extrapolation.

Par la prise en compte des phénomènes physiques de l'atmosphère, les approches par flot optique semblent plus adaptées pour décrire le mouvement des cellules nuageuses au cours du temps que les méthodes par association de blocs ou de cellules qui reposent sur une association entre deux images. C'est d'ailleurs la conclusion tirée de l'article [5] : ROVER et MOVA obtiennent de meilleures prévisions que TREC.

1.3 Méthodes par apprentissage profond

Nous avons vu, sans trop entrer dans les détails, différentes approches au problème "Rain Nowcasting". L'émergence des méthodes d'intelligence artificielle ont poussé des chercheurs à s'intéresser à l'application de celles-ci.

Différentes démarches par apprentissage profond ont été étudiées durant les dernières années : des réseaux de neurones récurrents, U-Net et réseaux antagonistes génératifs (GAN).

1.3.1 Approches récurrentes

ConvLSTM

L'observatoire de Hong Kong a développé le premier modèle d'apprentissage profond pour répondre à ce problème. Ses résultats sont comparés à la méthode ROVER. Cette dernière, bien qu'efficace possède de nombreux paramètres qui sont difficiles à déterminer pour donner de bonnes performances de prédiction. Par conséquent, des réseaux de neurones peuvent être capable d'apprendre à calculer le flot optique, comme le fait l'algorithme ROVER, sans avoir à définir différents paramètres au préalable.

La motivation qui a poussé les auteurs de ce papier à s'intéresser à un réseau récurrent convolutif provient de la recherche de travaux sur une problématique similaire. En effet, ce type d'architecture a été retenu pour la prédiction de la prochaine image dans une vidéo [6].

Dans [7], les auteurs se sont basés sur l'architecture récurrente *Long Short Term Memory* (LSTM) en ajoutant des opérations de convolutions pour garder l'information spatiale présente au sein des images. Ces couches se nomment ConvLSTM. Plusieurs couches ConvLSTM sont utilisées pour créer une architecture encodeur décodeur. Ainsi, les premières couches de l'encodeur vont permettre d'identifier des phénomènes plutôt locaux alors que les plus profondes vont détecter des mouvements plus globaux pour avoir une compréhension plus fine du phénomène météorologique que par une analyse à une seule échelle. Cela rejoint les approches par flot optique qui traitaient les images à plusieurs échelles suite aux limites des méthodes initiales comme TREC qui était restreint par la taille des blocs. L'information va ensuite être décodé pour réaliser la prédiction.

Le dataset choisi pour cette étude sont des données de réflectivité issues des radars météorologiques de Hong Kong. C'est donc un problème de régression pour l'estimation de la réflectivité (à noter qu'une relation relie la réflectivité au taux de précipitation). Ces données sont acquises toutes les 6 minutes et les séquences utilisées comportent 20 images (5 pour l'entraînement et 15 pour les prédictions). Les 97 jours les plus pluvieux ont été sélectionnés pour l'entraînement du modèle.

Différentes métriques sont utilisées pour évaluer les performances du modèle dont certaines se basant sur des valeurs binaires. Par conséquent, un seuil à un taux de précipitation de 0.5mm/h a été utilisé (valeur limite à laquelle il est considéré qu'il pleut).

Pour l'ensemble des métriques évoquées, les résultats obtenus par le réseau ConvLSTM sont meilleurs que l'approche par flot optique ROVER et que le réseau LSTM non convolutif. De même, la dégradation de la précision des prédictions diminue plus lentement pour le réseau récurrent convolutif.

Certaines remarques sont à tirer de cet article :

1. l'approche LSTM semble être adaptée par sa capacité à abandonner de l'information. Cependant, les méthodes par flot optique se basaient sur l'information des deux ou trois dernières images pour la prédiction des suivantes. Le réseau est alors peut-être trop complexe pour la tâche à réaliser.
2. la manière d'évaluer les performances du réseau ne permet pas d'estimer si ce dernier obtient de bonnes prédictions à taux de précipitation plus élevé. Or, ceci constituerait une information importante pour de nombreux secteurs d'activité météo-sensible.
3. la construction du dataset avec les 97 jours les plus pluvieux est discutable malgré la justification dans l'article que l'objectif du réseau est de prédire la pluie et que tous les jours ne sont pas pluvieux. Ce choix va biaiser le modèle lors de son utilisation opérationnelle.

TrajGRU

Les chercheurs ayant travaillé sur le modèle ConvLSTM, évoqué dans la partie précédente, ont planché sur une nouvelle approche après avoir identifié deux défauts majeurs :

1. la propriété d'invariance par localisation des filtres convolutifs utilisés qui ne permet pas de décrire fidèlement les phénomènes météorologiques. Les hyperparamètres des filtres convolutifs sont fixes (notamment la taille du noyau, le paramètre de dilatation). Par exemple, dans le cadre d'une rotation ou d'un changement d'échelle d'une cellule pluvieuse, la convolution aurait des difficultés pour rendre compte du phénomène comparé à une méthode basée sur le flot optique.
2. le dataset utilisé pour l'entraînement et la manière d'évaluer le modèle. En effet, ils reconnaissent que choisir les 97 jours les plus pluvieux comme données d'entraînement du modèle créé un biais. De plus, utiliser un seuil à 0.5 mm/h de pluie pour évaluer les performances du modèle n'est pas efficace pour rendre compte de son bon fonctionnement, notamment à haut taux de précipitation.

Partis de ce premier constat, ils ont proposé le modèle TrajGRU qui est convolutif et récurrent. Ce modèle utilise l'adaptation convolutive de la cellule *Gated Recurrent Unit* (GRU).

La particularité du modèle TrajGRU est d'apprendre les connections récurrentes de manière dynamique.

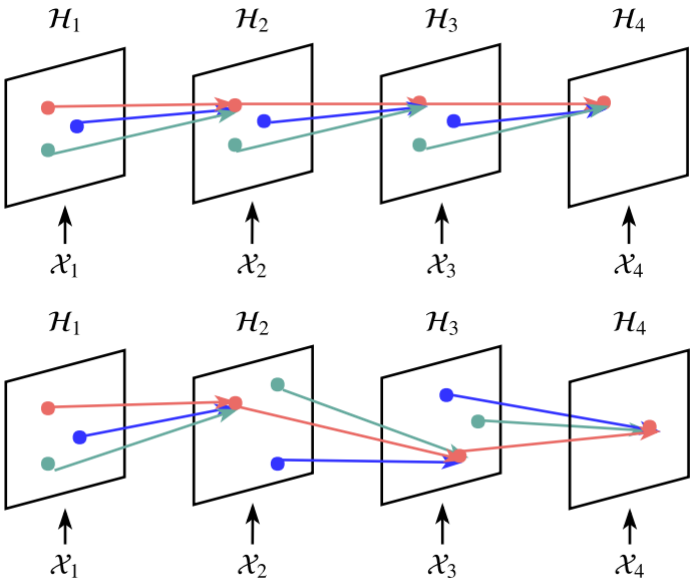

FIGURE 1.1 – Comparaison de la structure des connections entre un réseau convolutif récurrent classique (en haut) et le modèle TrajGRU (en bas). Images issues de [8].

Ainsi, pour apprendre de manière automatique les connections récurrentes, un réseau de neurones convolutif γ à une couche cachée a été utilisé. C'est donc un réseau très simple avec peu de paramètres. Il se base sur l'entrée à l'instant t X_t et l'état en mémoire à l'instant $t-1$ H_{t-1}. Les vecteurs de déplacement horizontaux et verticaux à l'instant t sont alors estimés comme : $U_t, V_t = \gamma(X_t, H_{t-1})$.

Le réseau utilisé dans ce papier est une structure encodeur-décodeur avec des cellules TrajGRU, du sous-échantillonage dans la partie encodeur et du sur-échantillonage dans le décodeur pour détecter des phénomènes à diverses échelles.

Sur la base du deuxième constat effectué, les auteurs ont changé la manière de créer le dataset. En effet, ils ont sélectionné l'ensemble des jours pluvieux (993 jours, ce qui est significativement plus que les 97 jours utilisés dans [7]). En effet, ils souhaitent utiliser le modèle de prévision du taux de précipitation seulement si un premier modèle a indiqué la veille s'il allait pleuvoir ou non.

Ils ont aussi pris en compte les limites de l'évaluation du modèle dans [7]. Premièrement, ils ont utilisé une erreur quadratique moyenne pondéré par un poids face au problème du déséquilibre du jeu de données vis-à-vis des différents niveaux de précipitations. En effet, la plupart des pixels du jeu de données correspondent à un taux de précipitation compris entre 0 et 0.5 mm/h indiquant pas de pluie ou très peu alors que seulement 1.56% ont un taux supérieur à 10 mm/h. De plus, plusieurs seuils de taux de précipitation ont été utilisés pour mieux rendre compte des performances du modèle lors l'évaluation avec des indicateurs binaires. Dans cet article, le problème est toujours de régression et les données sont des données de réflectivité. L'estimation du taux de précipitation se base sur la relation $Z - R$.

Enfin, il a été testé deux configurations différentes : un modèle entraîné normalement (qui, à partir de 5 images prédit les 20 suivantes) et un modèle entraîné auquel, lors de la phase de prédiction opérationnelle, on applique un réajustement des poids au fur

et à mesure que le modèle reçoit des séquences de 5 images (on qualifie de *online fine-tuning*). Cette deuxième méthode semble particulièrement adaptée opérationnellement car le modèle continue d'apprendre au fil du temps.

Les performances du modèle TrajGRU ont été comparées à d'autres modèles : l'approche ROVER basée sur le flot optique, des CNN classiques 2D et 3D (structure d'un Encodeur-Decodeur convolutionnel sans *skip-connections*), un modèle ConvGRU (même architecture que le TrajGRU sans les connections récurrentes dynamiques) et un modèle ConvGRU entrainé sans la fonction de perte pondérée pour équilibrer le jeu de données comme dans [7]. Le modèle TrajGRU performait mieux que l'ensemble des autres modèles selon l'ensemble des indicateurs et pour tous les taux de précipitations considérées. De plus, l'ensemble des modèles où le *online fine-tuning* a été appliqué a obtenu de meilleurs résultats par rapport aux mêmes modèles sans cet ajustement. Il est intéressant de noter que le modèle ConvGRU entraîné sans ajustement de la fonction de perte pondérée (similairement à [7] performait moins bien que l'approche par flot optique pour un taux de précipitation moyen ou élevé.

On peut émettre quelques remarques sur cet article :

1. cette article ne justifie pas le choix de l'utilisation d'un GRU convolutif vis-à-vis du modèle LSTM. On peut penser que c'est parce que le modèle GRU est légèrement plus simple que le modèle LSTM et qu'une telle complexité n'était pas nécessaire pour décrire le mouvement des cellules pluvieuses. Cette critique rejoint celle effectuée dans la partie précédente.
2. il n'est pas mentionné comment est entraîné le modèle γ qui est responsable de déterminer les connections récurrentes.
3. l'utilisation de jours seulement pluvieux pour la création du jeu de données est discutable. Certaines journées peuvent être pluvieuses avec un très faible taux de précipitations. Cependant, cela prend sens dans le cas de l'*online fine-tuning*, puisque trop de jours sans pluie pourraient faire dévier les performances du modèle, le rendant moins efficace à la prévision des précipitations.

1.3.2 Approches U-Net

Dans cette partie, nous allons nous intéresser aux réseaux U-Net. Plusieurs travaux se sont intéressés à l'utilisation de ce type de réseau pour répondre à la problématique de prévision de la précipitation.

Un premier article est particulièrement intéressant, puisqu'il construit un modèle U-Net et compare ses résultats à l'approche TrajGRU évoquée dans la partie précédente. L'entrée du réseau U-Net comporte 5 images radar de réflectivité consécutives. C'est un problème de régression.

Il montre que les résultats obtenus entre les deux modèles sont semblables : U-Net performe légèrement mieux à haut taux de précipitation que TrajGRU et inversement.

Cet article tire donc la conclusion que le U-Net utilisé, un modèle assez simple de part sa construction avec 4 couches convolutives dans l'encodeur et le décodeur, obtient

des résultats comparables à un modèle beaucoup plus complexe qui est le TrajGRU, particulièrement adapté à l'apprentissage de séries temporelles. [9]

Par conséquent, cela pose la question de l'intérêt de l'utilisation de modèle récurrent, ces derniers ne sont peut être pas si adaptés pour décrire ce phénomène météorologique.

Il faut tout de même noter que l'étude développée dans [9] ne s'intéressait qu'aux précipitations dans les 30 prochaines minutes (et donc à la prédiction de 5 images). Nous n'avons donc pas idée des performances à plus long terme.

L'article [10] se propose d'ajouter les données de vent en entrée du réseau de neurones en plus des données de précipitations. Contrairement à l'ensemble des articles évoqués précédemment qui se basent sur des images radar de réflectivité, le cumul des précipitations sur 5 minutes est utilisé. La sortie du modèle correspond aux cartes de prédictions des cumuls de précipitation.

Le problème est de classification. 3 seuils sont utlisés pour caractériser l'intensité de la pluie : 0.1 mm/h, 1 mm/h, 2.5 mm/h.

Les résultats montrent que le modèle intégrant les données de vent apportent de meilleurs résultats que le même modèle avec seulement les images de pluie. Par conséquent, on peut déduire de cet article que la prise en compte de données de vent apportent une réelle plus-value pour la qualité des prédictions.

1.3.3 Approche par réseau de neurones génératif

L'utilisation d'un réseau GAN développé dans [11] repose principalement sur une idée : bien que les modèles d'apprentissage profond prédisent avec précision les précipitations de faible intensité, leur utilité opérationnelle est encore limitée car ils manquent de contraintes. Par conséquent, ils ont tendance à produire des prévisions floues à des délais plus longs, entraînant de mauvaises performances pour des évènements de prévision de pluie à moyenne ou haute intensité puisque ces derniers sont les plus rares. Ainsi, l'utilisation de réseau de neurones génératifs va ajouter, de par l'utilisation de discriminateur, des contraintes supplémentaires forçant le modèle à produire des images de prévisions plus ressemblantes de la réalité et donc plus fiables à long terme.

Le réseau retenu est constitué de deux discriminateurs et d'un générateur. Un discriminateur spatial est un réseau convolutif qui se charge de distinguer, de manière individuelle, des observations radar de celles qui sont générées ce qui impose la cohérence spatiale et limite la production de prévisions dont les contours sont flous. Un discriminateur temporel, qui un réseau convolutif 3D, tente de discerner les séquences observées des séquences générées, imposant une cohérence temporelle et pénalisant les prédictions sautillantes (ayant une grande variation temporelle d'une image à l'autre). Le modèle générateur a la forme d'un encodeur décodeur, comme l'ensemble des autres modèles utilisés. Il utilise des cellules récurrentes ConvGRU. Le problème est de régression.

Les performances du réseau GAN sont comparées notamment à un réseau U-Net. A faible taux de précipitation, le réseau U-Net obtient de légers meilleurs résultats alors que le réseau GAN semble plus adapté pour des taux de précipitations plus élevés. Enfin,

sur l'observation qualitative des résultats de manière visuelle, les contours des zones de précipitations sont plus fidèles à ce que l'on peut observer dans la réalité. En effet, le U-Net a tendance à produire des prédictions floues avec le temps. Par exemple, pour des prévisions à 90 minutes, la résolution des images produites est inférieure à 1 km^2 alors que celles produites par le U-Net ont une résolution de 32 km^2. Il est important d'avoir des images de bonnes résolutions notamment dans le cadre de précipitations très intenses qui sont souvent localisées.

Chapitre 2

Rapport de fin de projet

2.1 Introduction

Ce rapport a pour but de présenter les travaux réalisés et les résultats obtenus dans le cadre du projet PRAT Rain Nowcasting. Il s'inscrit dans la suite du rapport bibliographique qui présente le problème à résoudre (l'estimation du taux de précipitation à court terme) et différentes approches de l'état de l'art. Il comprenait des méthodes dites traditionnelles utilisant l'estimation du flot optique et des nouvelles méthodes basées sur de l'apprentissage profond.

L'objectif de ce projet était d'expérimenter des approches de réseaux de neurones convolutifs récurrents issues de cet état de l'art pour en évaluer la pertinence et la plus-value face à l'approche U-Net [10]. Je me suis concentré sur le réseau de neurones TrajGRU développé dans [8] qui a pour spécificité d'apprendre les connexions récurrentes entre les couches cachées de manière automatique par l'approximation du flot optique. L'évaluation et la comparaison de ces approches on été effectuées sur des données radar de Météo-France de la région Brestoise.

2.2 Jeu de données

Les données utilisées au cours de ce projet sont les données radar de pluies et les données de vent acquises toutes les 5 minutes dans la région de Brest entre 2016 et 2018 inclus. [12]

2.2.1 Données de pluie

Les données de pluie sont des cartes de précipitations cumulatives sur 5 minutes de taille 128×128 pixels. Chaque pixel correspond donc à la quantité de pluie tombée sur les 5 dernières minutes en centième de millimètre ($10^{-2}mm$). Pour l'entraînement du réseau de neurones, ces cartes de pluies ont été converties en mm/h ce qui revient à multiplier chacun des pixels de ces cartes par un facteur $\frac{100}{12}$.

Il a été considéré 3 seuils de pluie $s_1 = 0.1$ mm/h, $s_2 = 1$ mm/h et $s_3 = 2.5$ mm/h de telle sorte que la pluie en un pixel de l'image, notée $p_{i,j}$, soit considérée comme :

Classe	Pixels avec cette classe	Images contenant cette classe
$0.1 \leq p_{i,j} < 1$	7.4%	61%
$1 \leq p_{i,j} < 2.5$	2.9%	43%
$2.5 \leq p_{i,j}$	1.2%	34%

TABLE 2.1 – Distribution des classes du taux de précipitation au sein du jeu de données [10]

- inexistante si 0 mm/h $\leq p_{i,j} < 0.1$ mm/h
- faible si 0.1 mm/h $\leq p_{i,j} < 1$ mm/h
- modérée si 1 mm/h $\leq p_{i,j} < 2.5$ mm/h
- forte si 2.5 mm/h $\leq p_{i,j}$

Le jeu de données est assez déséquilibré, les pluies les plus fortes apparaissant de manière moins fréquentes comme en témoigne le tableau n°1.

Pour pallier ce déséquilibre du jeu de données, nous allons utiliser une fonction de perte pondérée en fonction des classes qui sera détaillée dans la partie 2.3.2.

2.2.2 Données de vent

Comme les cartes de pluie, les données de vent sont acquises toutes les 5 minutes sous la forme de deux images de vent U et V qui correspondent à la composante de vitesse du vent respectivement de l'Ouest vers l'Est et du Sud vers le Nord. Chaque pixel de ces cartes renseignent la vitesse du vent en m/s à un instant t donné dans une direction.

J'ai décidé d'appliquer une standardisation aux données de vent qui a pour effet de centrer et réduire les composantes de vent. Celle ci avait déjà été réalisée dans [10]

$$U_{i,j,t} \leftarrow \frac{U_{i,j,t} - \mu_U}{\sigma_U}$$
$$V_{i,j,t} \leftarrow \frac{V_{i,j,t} - \mu_V}{\sigma_V}$$

où i, j correspondent aux indices du pixel traité et t correspond à date d'acquisition de la carte de vent.

2.2.3 Construction des séquences

Les entrées du réseau de neurones correspondent à l'accumulation d'une heure de données de pluie par la concaténation de 12 cartes successives. Dans le cas où nous souhaitons également utiliser le vent, le même procédé peut être réalisé pour chaque composante de vent. L'ensemble des données (pluie, composante U et V de vent) peuvent être concaténées entre elles et passées en entrée du réseau de neurones.

A chaque entrée est associée une vérité terrain qui joue le rôle d'objectif de prédiction pour notre réseau de neurones. Cette vérité terrain se définit comme la concaténation des 12 cartes de pluie acquises successivement après la dernière donnée en entrée. Les réseaux de neurones ont donc pour objectif d'estimer des cartes de pluie à intervalles de temps régulier de 5 minutes pour un horizon d'une heure.

Cette approche est légèrement différente de celle qui a été choisie dans [10], puisqu'un réseau ne produisait qu'une seule carte de pluie à 30 ou 60 minutes. Cependant, dans le cadre d'approche récurrente, qui utilise un principe de mémoire des états, le modèle ne peut prédire des états que de manière successive. Ainsi, pour émettre une prédiction à $t + 30$ minutes (avec t la date de la dernière entrée du réseau), des estimations à $t + 5$, $t + 10$, ... , $t + 25$ devront être effectuées. C'est pour cette raison que j'ai décidé de créer une séquence objectif du réseau de neurones composée de 12 cartes de pluie successives.

Par conséquent, une donnée correspond à une séquence de 12 cartes de pluie en entrée et ses potentielles 24 cartes de vent associées ainsi que des 12 cartes de pluie qui correspondent à la vérité terrain permettant d'évaluer les résultats de l'algorithme.

2.2.4 Entraînement, validation et test

La base de données a été coupée en 3 jeux de données pour la phase d'entraînement, de validation et de test. Le jeu d'entraînement est constitué des données des années 2016 et 2017. L'année 2018 est réservée aux jeux de validation et de test. Une semaine sur deux appartient au jeu de validation et l'autre au jeu de test pour éviter de biaiser l'évaluation étant donné que la pluie dépend des saisons.

Plusieurs choix ont été fait sur la manière de créer les séquences :

1. Les séquences en entrée ne se recoupent pas d'une donnée à l'autre. Cela signifie qu'une carte de pluie et de vent n'appartient qu'à une unique séquence d'entrée pour éviter le sur-apprentissage.
2. Si une donnée présente une carte de pluie en entrée ou en sortie qui contient un ou plusieurs pixels de pluie égal à -1 (correspondant à une erreur d'acquisition), la donnée est supprimée.
3. Si la dernière image de la séquence d'entrée ne contient pas de pluie, la donnée ne sera également pas considérée. En effet, une grande partie de notre jeu de données ne contient pas de pluie. J'élimine ces séquences car elles risquent de faire diverger le réseau de neurones de ce pourquoi il est entraîné : estimer le taux de précipitation.

Cette procédure a été appliquée à chacun des jeux de données.

2.3 Réalisation du projet

2.3.1 Modèles utilisés

Dans un premier temps, pour prendre en main les données et le problème, j'ai décidé d'implémenter un réseau de neurones, entièrement convolutif, sous la forme d'un encodeur

décodeur. Il se compose de couches convolutives dans la partie encodeur et de couches convolutives transposées dans la partie décodeur. Ce réseau est inspiré du réseau CNN_2D dans [8].

J'ai ensuite utilisé le réseau convolutif récurrent TrajGRU [8] pour évaluer ses performances sur notre jeu de données. Pour rappel, cette couche est une extension de la cellule ConvGRU où les connexions récurrentes entre les états cachés sont apprises de manière automatique par le réseau.

Pour évaluer la pertinence de ce module de trajectoire, j'ai enfin comparé ces résultats à un réseau ConvGRU.

Un encodeur décodeur convolutif (CNN_2D)

Les résultats obtenus avec ce réseau m'ont servi de ligne de base pour les autres méthodes. Son architecture est assez simple et est illustrée sur la figure n°2.1. Il consiste en un encodeur-décodeur où chaque couche convolutive et convolutive transposée, hormis celle de sortie, est suivie d'une couche de BatchNormalization et d'une fonction d'activation ReLU. Les couches convolutives et convolutives transposées ont un pas de 2 pour permettre de, respectivement, sous-échantillonner et sur-échantillonner l'image.

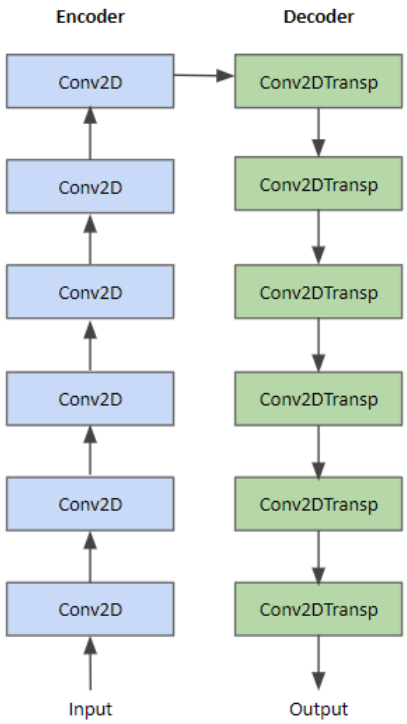

FIGURE 2.1 – Schéma du réseau de neurones convolutifs simple.

Le tableau n°3 en annexe détaille l'architecture de ce réseau.

ConvGRU

L'architecture est illustrée sur la figure n°2.2. Ce réseau a une architecture similaire au réseau précédent puisqu'il consiste en un encodeur-décodeur. La partie encodeur est constituée d'une alternance de 3 couches convolutives, sous-échantillonnant l'image, et 3 couches récurrentes ConvGRU. Chaque couche convolutive est suivie d'une fonction d'activation ReLU. La partie décodeur est symétrique à la partie encodeur en remplaçant les couches convolutives par des couches convolutives transposées qui ont pour effet de sur-échantillonner l'image. Une opération de convolution est appliquée en sortie pour transformer le résultat.

Contrairement au réseau précédent, l'utilisation de cellules récurrentes amène à traiter le problème différemment. Le réseau de neurones reçoit de manière séquentielle les données et les états des couches récurrentes correspondants à une même résolution sont transmises d'un pas de temps à un autre.

La formulation des opérations effectuées dans une cellule ConvGRU est décrite par le système d'équation n°1 où $\mathcal{Z}_t$, $\mathcal{R}_t$, $\mathcal{H}'_t$ et $\mathcal{H}_t$ correspondent respectivement à la porte de mise à jour, de réinitialisation, de nouvelle information et à l'état de mémoire à l'instant t. $\mathcal{X}_t$ correspond à l'entrée à l'instant t. La fonction σ est la fonction sigmoid, utilisée comme fonction d'activation, alors que la fonction f correspond à une fonction d'activation LeakyReLU. $\mathcal{W}$ correspond au tenseur de poids de la cellule. [8]

$$\begin{cases} \mathcal{Z}_t & = \sigma\left(\mathcal{W}_{xz} * \mathcal{X}_t + \mathcal{W}_{hz} * \mathcal{H}_{t-1}\right) \\ \mathcal{R}_t & = \sigma\left(\mathcal{W}_{xr} * \mathcal{X}_t + \mathcal{W}_{hr} * \mathcal{H}_{t-1}\right) \\ \mathcal{H}'_t & = f\left(\mathcal{W}_{xh} * \mathcal{X}_t + \mathcal{R}_t \circ \left(\mathcal{W}_{hh} * \mathcal{H}_{t-1}\right)\right) \\ \mathcal{H}_t & = \left(1 - \mathcal{Z}_t\right) \circ \mathcal{H}'_t + \mathcal{Z}_t \circ \mathcal{H}_{t-1} \end{cases} \tag{2.1}$$

Le tableau n°4 en annexe détaille l'architecture de ce réseau.

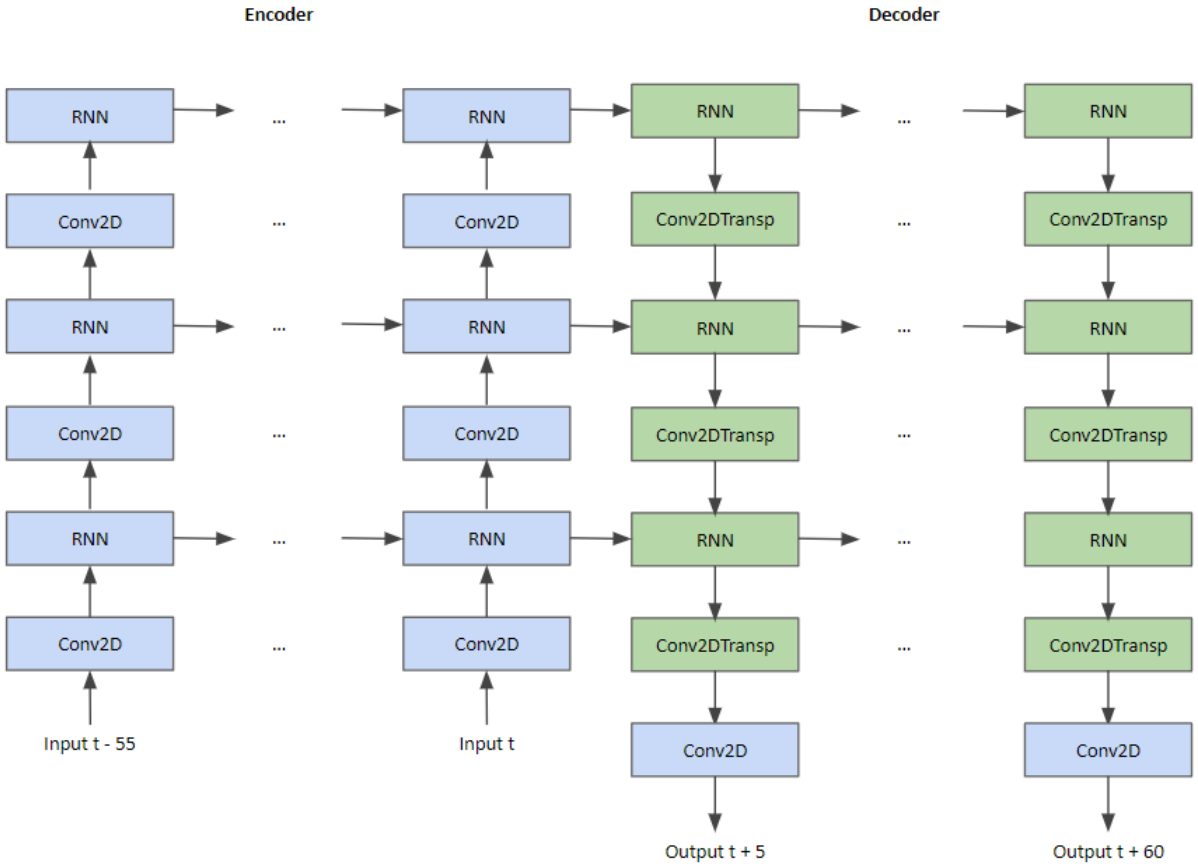

FIGURE 2.2 – Schéma des réseaux de neurones convolutifs récurrents utilisant des cellules ConvGRU et TrajGRU.

TrajGRU

L'architecture de ce réseau de neurones est la même que le réseau de neurones ConvGRU précédent que l'on peut visualiser sur la Figure n°2.2. Seul le type de cellule récurrente utilisée change. La cellule récurrente TrajGRU est une extension de la cellule ConvGRU dans laquelle les connexions récurrentes entre les états cachés sont apprises de manière automatique.

Le système d'équation n°2 décrit le fonctionnement de la cellule TrajGRU. Les notations déjà utilisées pour le système d'équation n°1 sont les mêmes. γ est un réseau de neurones qui estime les composantes U_t et V_t du flot optique à partir de l'entrée $\mathcal{X}_t$ à l'instant t et l'état $\mathcal{H}_{t-1}$ à l'instant $t-1$. Il consiste en un réseau d'une couche cachée convolutive ayant un noyau de taille 5×5 et 32 canaux. Il est intégré au sein de la couche TrajGRU. L est le nombre de liens autorisés et warp $(\mathcal{H}_{t-1}, \mathcal{U}_{t,l}, \mathcal{V}_{t,l})$ est une fonction qui sélectionne les positions pointés par $\mathcal{U}_{t,l}, \mathcal{V}_{t,l}$ dans $\mathcal{H}_{t-1}$. [8]

$$
\begin{cases}
\mathcal{U}_t, \mathcal{V}_t & = \gamma\left(\mathcal{X}_t, \mathcal{H}_{t-1}\right), \\
\mathcal{Z}_t & = \sigma\left(\mathcal{W}_{xz} * \mathcal{X}_t + \sum_{l=1}^{L} \mathcal{W}_{hz}^{l} * \operatorname{warp}\left(\mathcal{H}_{t-1}, \mathcal{U}_{t,l}, \mathcal{V}_{t,l}\right)\right), \\
\mathcal{R}_t & = \sigma\left(\mathcal{W}_{xr} * \mathcal{X}_t + \sum_{l=1}^{L} \mathcal{W}_{hr}^{l} * \operatorname{warp}\left(\mathcal{H}_{t-1}, \mathcal{U}_{t,l}, \mathcal{V}_{t,l}\right)\right), \\
\mathcal{H}'_t & = f\left(\mathcal{W}_{xh} * \mathcal{X}_t + \mathcal{R}_t \circ \left(\sum_{l=1}^{L} \mathcal{W}_{hh}^{l} * \operatorname{warp}\left(\mathcal{H}_{t-1}, \mathcal{U}_{t,l}, \mathcal{V}_{t,l}\right)\right)\right), \\
\mathcal{H}_t & = (1 - \mathcal{Z}_t) \circ \mathcal{H}'_t + \mathcal{Z}_t \circ \mathcal{H}_{t-1}.
\end{cases}
\tag{2.2}
$$

Le tableau n°5 en annexe détaille l'architecture de ce réseau.

2.3.2 Entraînement et évaluation des modèles

Paramètres d'entraînement

L'ensemble des réseaux de neurones ont été entraîné avec une fonction de perte pondérée en fonction du taux de précipitation. En effet, nous avons vu que le jeu de données est déséquilibré et contenait moins de données à fort taux de précipitation qu'à un plus faible taux.

La fonction de perte choisie est la suivante :

$$
f_{loss}(y, \hat{y}) = \frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{128} \sum_{j=1}^{128} w_{n,i,j}((y_{n,i,j} - \hat{y_{n,i,j}})^2 + |y_{n,i,j} - \hat{y_{n,i,j}}|)
$$

avec $y_{n,i,j}$ qui au taux de précipitation du pixel (i, j) de la n-ième image de la vérité terrain, $\hat{y_{n,i,j}}$ est l'estimation de $y_{n,i,j}$ faite par le réseau de neurones. $w_{n,i,j}$ est le poids qui correspond à la valeur de la vérité terrain $y_{n,i,j}$. N correspond au nombre d'image. En pratique, j'ai appliqué un facteur de 5×10^{-4} à cette fonction de perte pour simplement avoir une valeur de perte plus petite (de l'ordre de la centaine).

La fonction w se définit de la manière suivante :

$$
w(x) = \begin{cases}
1, & x < 0.1 \\
1.5, & 0.1 \leq x < 1 \\
2, & 1 \leq x < 2.5 \\
3, & 2.5 \leq x
\end{cases}
$$

Elle a été choisie de manière empirique au vu de la distribution des données de pluie évoquée dans la partie 2. Elle a pour objectif de compenser l'effet du déséquilibre du jeu de données.

Les réseaux de neurones ont été entraînés avec l'optimiseur Adam avec $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$ et un batch size de 4. Les réseaux de neurones entraînés sans vent avaient un pas d'apprentissage de 10^{-5}, et ceux prenant en compte le vent en entrée ont été optimisés avec un pas d'apprentissage de 5×10^{-6}. Les réseaux ont été entrainés sur 75 epochs sur un GPU Tesla V100-SXM2-32GB.

Métriques utilisées pour l'évaluation

La performance des modèles est évaluée avec 3 métriques aussi utilisées dans [10]. Cela permet de comparer les résultats avec ceux obtenus par le réseau U-Net. Notre problème

étant initialement de régression, il est appliqué des seuils à différents taux de précipitation : à 0.1 mm/h, 1 mm/h et 2.5 mm/h.

La première métrique utilisée est le F1-Score pour la classe m qui se définit de la manière suivante :

$$F1_m = 2\frac{\text{Précision }_m \times \text{ Rappel }_m}{\text{Précision }_m + \text{ Rappel }_m}$$

où la précision et le rappel sont :

$$\text{Précision }_m = \frac{\sum_{n=1}^{N}\sum_{i=1}^{128}\sum_{j=1}^{128} TP_{n,i,j}^m}{\sum_{n=1}^{N}\sum_{i=1}^{128}\sum_{j=1}^{128} TP_{n,i,j}^m + FP_{n,i,j}^m}$$

$$\text{Rappel }_m = \frac{\sum_{n=1}^{N}\sum_{i=1}^{128}\sum_{j=1}^{128} TP_{n,i,j}^m}{\sum_{n=1}^{N}\sum_{i=1}^{128}\sum_{j=1}^{128} TP_{n,i,j}^m + FN_{n,i,j}^m}$$

où :

- $TP_{n,i,j}^m = 1$ si la classe estimée au pixel i, j est de l'image n est m et la classe du pixel de la vérité terrain correspondant est m.
- $FP_{n,i,j}^m = 1$ si la classe estimée au pixel i, j est de l'image n est m et la classe du pixel de la vérité terrain correspondant est différent de m.
- $FN_{n,i,j}^m = 1$ si la classe estimée au pixel i, j est de l'image n n'est pas m et la classe du pixel de la vérité terrain correspondant est m.

Ces scores sont compris entre 0 et 1, où 0 correspond au pire classifieur et 1 au meilleur pour la classe m.

La seconde métrique utilisée est le Threat Score (TS) qui se définit :

$$TS_m = \frac{\sum_{n=1}^{N}\sum_{i=1}^{128}\sum_{j=1}^{128} TP_{n,i,j}^m}{\sum_{n=1}^{N}\sum_{i=1}^{128}\sum_{j=1}^{128} TP_{n,i,j}^m + FP_{n,i,j}^m + FN_{n,i,j}^m}$$

C'est aussi un indicateur entre 0 et 1, où 0 correspond au pire classifieur et 1 au meilleur pour la classe m. Ce score caractérise le taux de réussite à la bonne classification de données dans la classe m.

Enfin, le biais pour la classe m est le suivant :

$$BIAS_m = \frac{\sum_{n=1}^{N}\sum_{i=1}^{128}\sum_{j=1}^{128} TP_{n,i,j}^m + FP_{n,i,j}^m}{\sum_{n=1}^{N}\sum_{i=1}^{128}\sum_{j=1}^{128} TP_{n,i,j}^m + FN_{n,i,j}^m}$$

Il permet de rendre compte à quel point le classifieur est biaisé pour la classe m. Un score de 1 correspond à un classifieur parfait, un score supérieur à 1 correspond à un classifieur qui a tendance à surestimer la pluie vis-à-vis de la classe m, alors qu'un score inférieur à 1 signifie que le réseau sous estime la pluie pour cette classe.

2.3.3 Résultats quantitatifs obtenus

Temps	Model	F1-Score			TS-Score			Bias-Score		
		Class 1	Class 2	Class 3	Class 1	Class 2	Class 3	Class 1	Class 2	Class 3
30 min	**Persistance**	0.602	0.438	0.264	0.43	0.28	0.152	**0.993**	**0.999**	**1.006**
	CNN 2D	0.647	0.522	0.386	0.478	0.353	0.239	1.227	1.705	1.404
	ConvGRU	0.689	0.592	0.425	0.525	0.42	0.27	1.104	1.088	0.755
	TrajGRU	0.685	**0.604**	0.461	0.521	**0.433**	0.3	1.129	1.196	0.904
	ConvGRU_Wind	0.674	0.595	0.476	0.508	0.424	0.313	1.224	1.342	1.211
	TrajGRU_Wind	0.667	0.598	**0.48**	0.501	0.426	**0.316**	1.297	1.245	1.021
	UNet_Wind	**0.724**	0.581	0.43	**0.567**	0.41	0.374	0.907	0.817	0.699
60 min	**Persistance**	0.523	0.338	0.166	0.354	0.203	0.09	**0.998**	**0.997**	**0.997**
	CNN 2D	0.574	0.441	0.249	0.403	0.283	0.142	1.252	1.455	0.955
	ConvGRU	0.596	0.456	0.171	0.425	0.296	0.093	1.017	0.818	0.293
	TrajGRU	**0.598**	0.493	0.274	**0.426**	0.327	0.159	1.215	1.099	0.547
	ConvGRU_Wind	0.581	0.485	0.311	0.41	0.32	0.184	1.468	1.592	1.281
	TrajGRU_Wind	0.578	**0.502**	**0.319**	0.407	**0.335**	**0.19**	1.53	1.303	0.819
	UNet_Wind	0.569	0.391	0.204	0.397	0.243	0.113	0.848	0.78	0.627

TABLE 2.2 – Comparaison des performances des réseaux sur les différentes métriques.

Le tableau n°2 présente les scores obtenus sur chaque métrique des différents réseaux de neurones sur le jeu de test.

La persistance donne un premier point de repère. Il consiste à utiliser la dernière image acquise comme prédiction. Le score obtenu pour le biais est donc, assez logiquement, particulièrement bon.

Le modèle convolutif simple, utilisé pour prendre en main le problème et obtenir un premier résultat, donne sur l'ensemble des métriques des résultats plus satisfaisants que la persistance. Il est en revanche moins efficace comparé à l'ensemble des réseaux de neurones récurrents testés.

Sans l'utilisation du vent, on constate que le réseau TrajGRU se comporte nettement mieux par rapport au ConvGRU au niveau des métriques pour les taux de précipitations modérés et forts. Ceci est encore plus marquant à $t + 60$ minutes. On peut donc penser que la prise en compte d'un module apprenant les connexions récurrentes de manière dynamique par l'estimation du flot optique aide à mieux comprendre le mouvement des cellules pluvieuses très actives donnant lieu à de fortes précipitations.

Avec l'utilisation des données de vent en entrée, on constate que la différence de résultats obtenus entre le ConvGRU et le TrajGRU est plus faible qu'auparavant ce qui montre bien l'utilité, dans le cas où nous n'utilisions pas le vent, d'apprendre de manière automatique les connexions récurrentes. Le TrajGRU performe tout de même légèrement mieux pour des pluies fortes et modérées, et plus particulièrement à $t + 60$ minutes.

On tire de ces résultats que les réseaux récurrents performent mieux que le réseau U-Net de [10] pour les pluies modérées et fortes. On peut penser que le suivi des cellules pluvieuses actives est meilleur avec l'utilisation d'information de récurrence au sein du modèle. Cette analyse est à prendre avec du recul, puisque la fonction de perte utilisée pour l'entraînement du U-Net est différente (une entropie croisée) ainsi que la façon de pallier le déséquilibre des données (sur-échantillonage de cartes de pluie).

Nous pouvons aussi penser que la façon d'entraîner le réseau ait un impact sur les résultats obtenus, notamment de manière plus marquée à $t + 60$ minutes. En effet, deux réseaux U-Net ont été entraînés, un prédisant les classes de pluies à $t + 30$ minutes et l'autre à $t + 60$ minutes. Or les autres réseaux présentés dans ce rapport prédisent le taux de précipitation de pluies toutes les 5 minutes pendant une heure. J'ai utilisé chacune de ces cartes prédites dans le calcul de la fonction de coût. Il est possible que cela aide le modèle à effectuer de meilleures prédictions une heure à l'avance. C'est d'ailleurs ce qu'on observe avec le réseau CNN_2D, qui obtient de meilleurs résultats par rapport au U_Net à plus long terme, alors que ce dernier est beaucoup plus précis pour $t+30$ minutes.

Il est à noter que pour chaque réseau de neurones, j'ai essayé d'en optimiser les hyperparamètres de manière assez empirique en faisant varier quelques paramètres tels que le *learning rate*, le *weight decay* ou encore en appliquant un *scheduler* pour réduire le *learning rate* au cours de l'entraînement bien que l'optimiseur Adam l'adapte déjà légèrement de manière automatique. Les entraînements étaient très sensibles aux hyperparamètres utilisés, notamment le pas d'apprentissage. Il aurait été judicieux de chercher à les optimiser avec une librairie telle que *Hyperopt* qui se charge de chercher un jeu de paramètres optimal pour maximiser les performances des modèles. Cependant, une epoch du réseau TrajGRU durant 1 heure, il n'a pas été possible d'utiliser cette solution dans le temps imparti du projet.

2.3.4 Résultats visuels

Dans cette partie, des exemples de prédictions réalisés par le réseau TrajGRU avec vent vont être montrés. Ces images ont été générées à partir du jeu de test. L'objectif est de montrer un panel d'illustrations rendant compte des avantages et inconvénients de ce modèle.

Les figures présentées afficheront :

- Sur la première ligne, les entrées aux temps $t-20$, $t-15$, $t-10$, $t-5$, t.
- Sur la seconde ligne, les prédictions réalisées aux temps $t+5$, $t+15$, $t+25$, $t+35$, $t+45$
- Sur la troisième ligne, les vérités terrains aux temps $t+5$, $t+15$, $t+25$, $t+35$, $t+45$

Un code couleur est utilisé pour rendre compte de l'intensité de la pluie :

- Un pixel est de couleur verte si le taux de précipitation est faible ($0.1 \leq p_{i,j} < 1$)
- Un pixel est de couleur bleue si le taux de précipitation est modéré ($1 \leq p_{i,j} < 2.5$)
- Un pixel est de couleur rouge si le taux de précipitation est fort ($2.5 \leq p_{i,j}$)

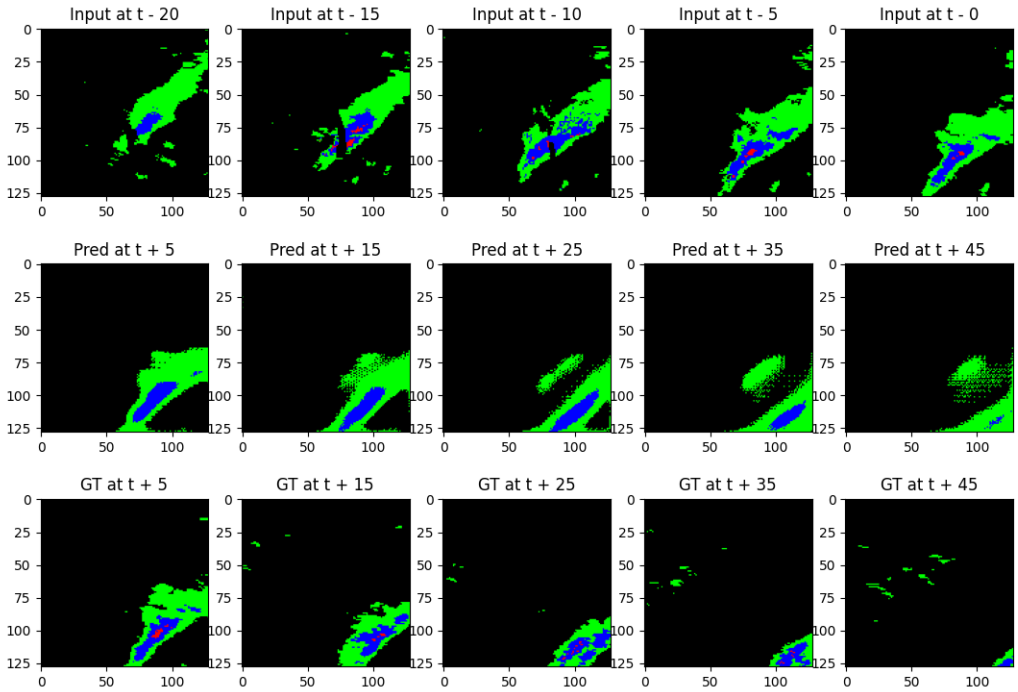

FIGURE 2.3 – Exemple de bonne estimation du mouvement de la cellule pluvieuse.

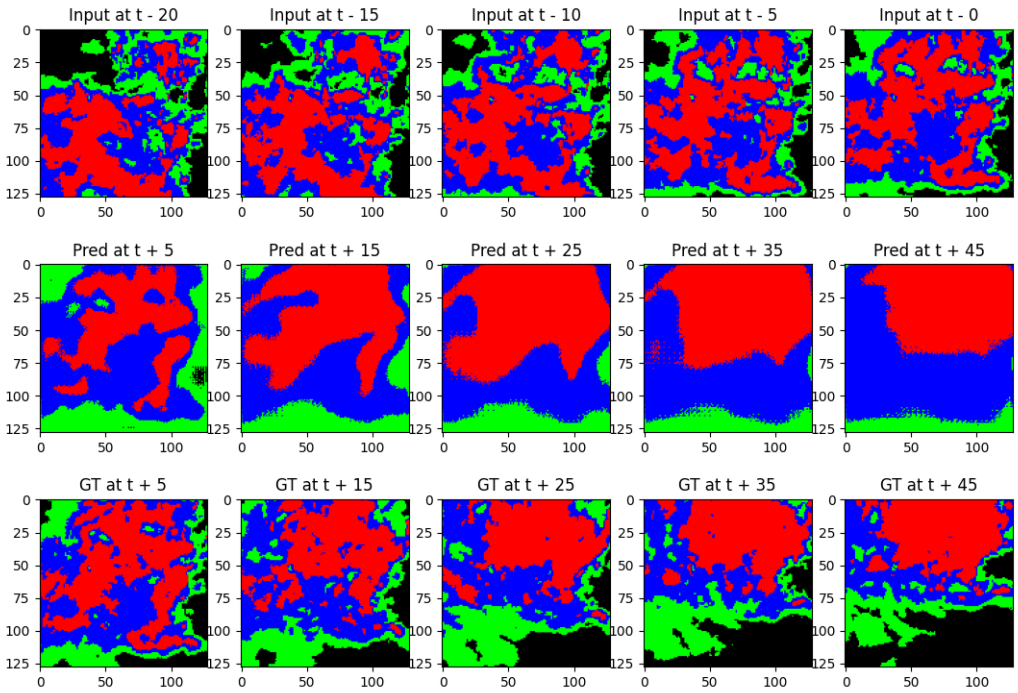

FIGURE 2.4 – Exemple de bon estimation de la zone pluvieuse à forte intensité.

Les figures n°2.3 et n°2.4 présentent des cas de plutôt bonnes prédictions étant donné la difficulté du problème. Le mouvement estimé de la cellule pluvieuse de la première prédiction est en accord avec la vérité terrain même si on observe un artefact avec une petite zone de pluie qui se détache de la cellule active. Le mouvement de la zone la plus pluvieuse de la seconde prédiction est également assez réaliste. Pour les autres classes, le modèle semble surestimer le taux de précipitation. Un autre constat qui revient sur l'ensemble des prédictions peut également être tiré : le réseau de neurones a tendance à lisser les bords des zones pluvieuses de différentes intensités.

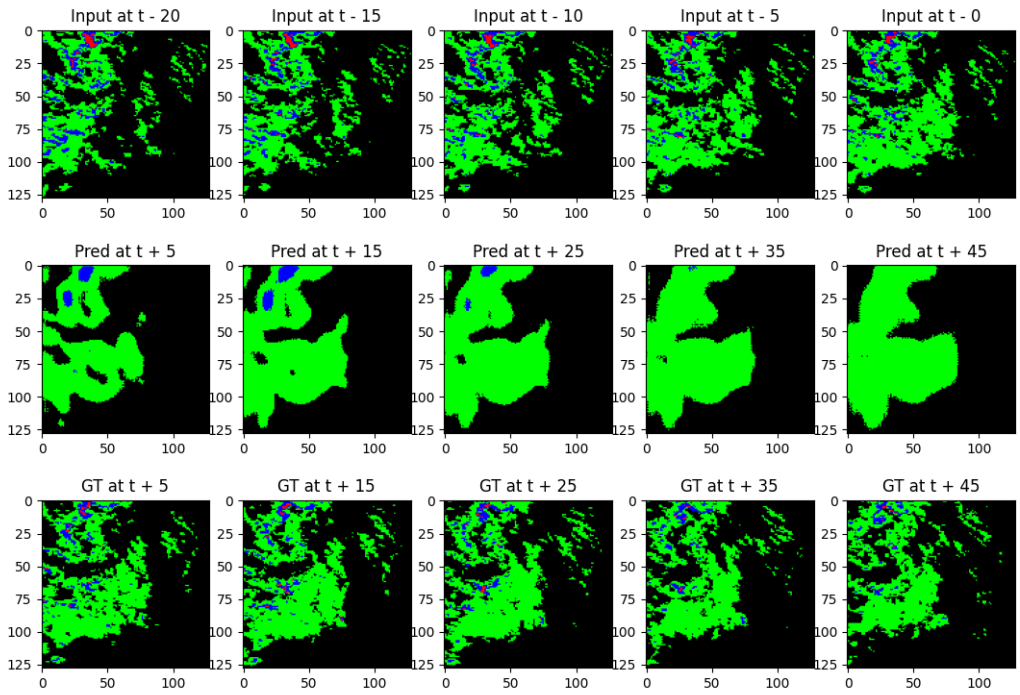

FIGURE 2.5 – Exemple de bon estimation de la zone pluvieuse à forte intensité.

La figure n°2.5 est une assez bonne prédiction. Cet exemple montre que dans le cas de vérité terrain avec quelques zones clairsemées, les prévisions réalisées sont en revanche plus denses. Le modèle peine à estimer les petites zones à plus fort taux de précipitation.

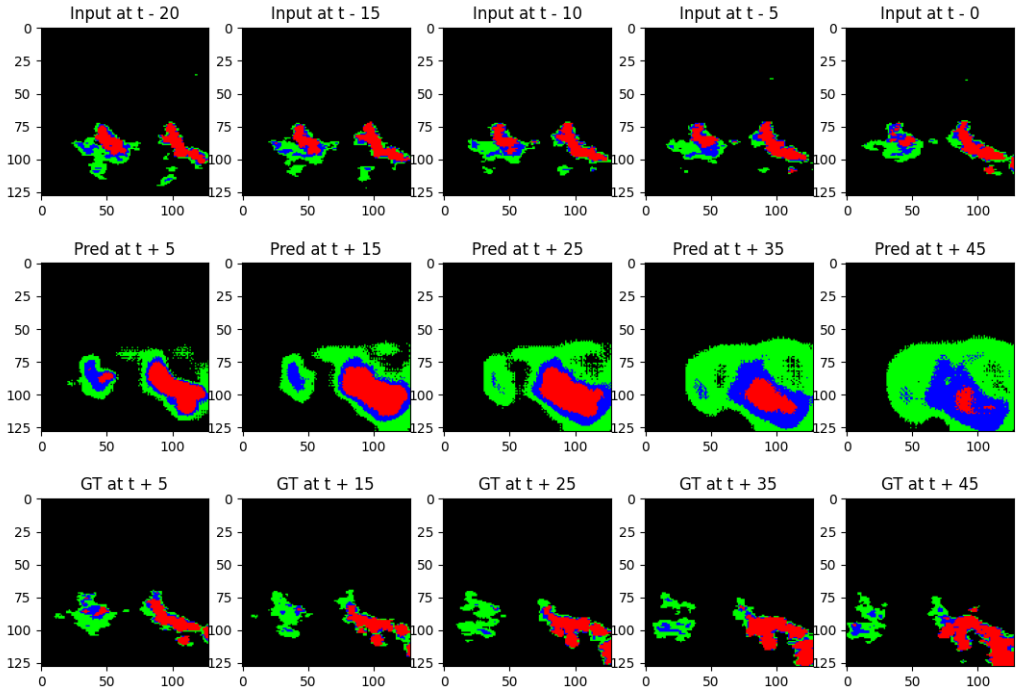

FIGURE 2.6 – Limite intervenant dans les cas de zones pluvieuses localisées de forte intensité.

La figure n°2.6 permet de rendre compte de plusieurs problèmes survenant avec notre

modèle.

Premièrement, elle montre l'exemple d'une zone pluvieuse de forte intensité très localisée. Cependant, dans le jeu de données, ce genre de cas est assez rare. En effet, les cellules très pluvieuses sont généralement au centre d'une grosse perturbation et sont donc entourées de zones de pluies de plus faible intensité. Ainsi, c'est ce à quoi le modèle est habitué à être confronté et prédit ainsi une région de pluie plus éparse que la réalité, avec des zones de pluie moins intenses.

Deuxièmement, au vu des données en entrée affiché, il ne semble pas y avoir de mouvement apparent. Ainsi, n'ayant probablement pas d'indication de comment les cellules pluvieuses vont bouger dans le temps, le modèle fait grossir la zone pluvieuse de faible intensité dans l'ensemble des directions dans le but d'essayer de minimiser l'erreur engendrée.

Enfin, cet exemple permet également de rendre compte que plus la prédiction est lointaine et plus le modèle a tendance à sous-estimer les pluies de fortes intensités. Ceci est un constat qui revient également dans plusieurs exemples de prédictions réalisées.

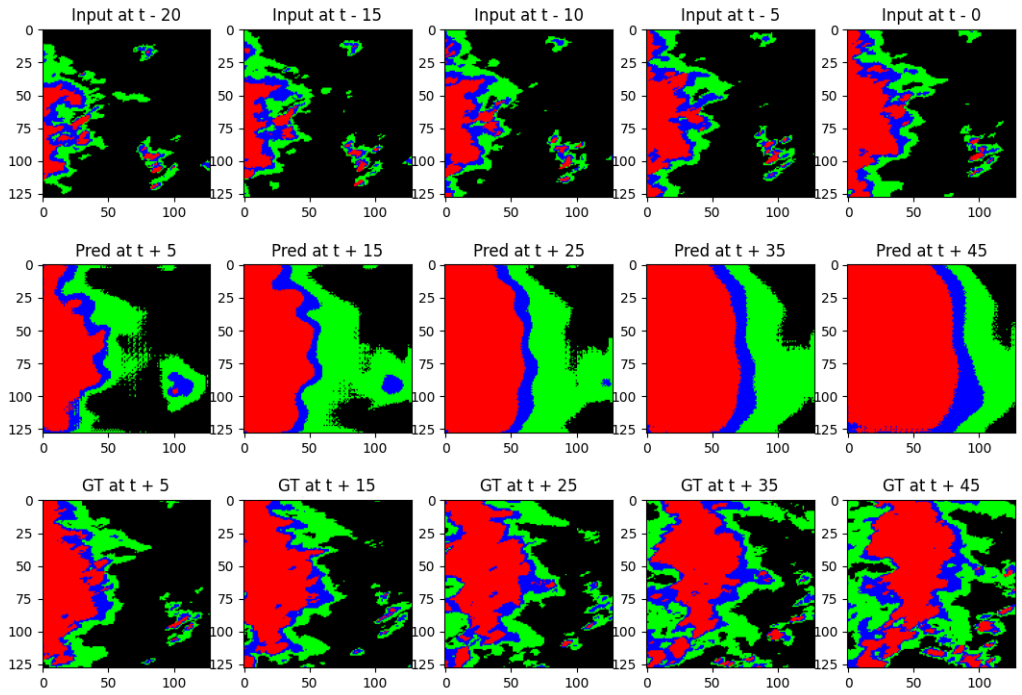

FIGURE 2.7 – Cas où l'information est en dehors du champ de vision.

La figure n°2.7 montre un cas limite de la fenêtre d'étude. En effet, la cellule pluvieuse bouge dans toute sa globalité de gauche à droite avec une pluie de forte intensité. Ainsi, le réseau est obligé d'extrapoler pour essayer d'estimer l'information et par conséquent surestime la pluie. Cependant, ce problème n'est pas spécifique au modèle utilisé mais plus à la formulation du problème.

A l'inverse et de manière assez logique, lorsqu'une cellule pluvieuse unique vient en dehors du champ de vision, le modèle ne peut la détecter.

2.4 Conclusion et points d'amélioration

A travers ce projet, nous avons pu voir l'intérêt de l'utilisation de réseaux récurrents pour le problème de prévision du taux de précipitation à court terme. Sans l'utilisation du vent, une architecture comprenant des couches récurrentes TrajGRU permettait d'obtenir de meilleurs résultats, plus particulièrement pour des taux de précipitation modérés et élevés, grâce au module d'estimation du flot optique qui permet d'apprendre les connexions récurrentes entre les états cachés de manière dynamique. Ce module de trajectoire était, assez logiquement, moins utile avec la présence au préalable des données de vent. Néanmoins, il faut rester prudent quant à l'interprétation des résultats de cette étude, le réseau U-Net a été entraîné différemment des réseaux convolutifs récurrents. En effet, il était appliqué à un problème de classification en fonction des seuils de pluie considérés dans cet article, avec des paramètres différents d'entraînements notamment au niveau de la fonction de perte et de la manière de rééquilibrer le jeu de données.

Nous pourrions envisager plusieurs pistes pour améliorer les résultats.

Premièrement, nous avons vu que l'entraînement était très sensible aux hyperparamètres. Cependant, ce dernier étant également très long (plusieurs jours), je n'ai pas eu le temps d'explorer un très vaste champ d'hyperparamètres possibles. Une solution serait d'utiliser une librairie telle que *Hyperopt* qui permet de trouver des paramètres proches de l'optimalité.

Deuxièmement, il pourrait être intéressant d'étudier une architecture TrajLSTM. Si ce projet nous a permis de montrer que l'utilisation d'information de récurrence était pertinent, il pourrait être judicieux de construire une architecture LSTM convolutive, modélisant mieux les dépendances à long terme, en y ajoutant le module de trajectoire. Cependant, il est très probable qu'une architecture GRU convolutive soit suffisante pour modéliser la dimension temporelle du problème. D'un point de vue purement visuel, les dernières images acquises semblent suffire à prédire les images de pluie suivantes.

Nous pouvons également essayer de complexifier l'architecture utilisée par l'ajout d'un ou de plusieurs autres niveaux de couches dans les parties encodeurs et décodeurs. Il est à noter qu'une couche récurrente convolutive est très consommatrice en mémoire RAM.

Pour finir, l'architecture GAN [11] parue récemment dans la littérature semble être une alternative intéressante pour obtenir des cartes de pluie en sortie plus fidèles à la réalité. Son générateur se base sur une architecture très similaire à celle du ConvGRU ou TrajGRU utilisée dans ce projet : c'est un encodeur décodeur utilisant des couches récurrentes convolutives. La différence se joue au niveau du discriminateur temporel et spatial qui ajoute des contraintes sur le réalisme des cartes de pluie générées. Cela pourrait être une solution à certains problèmes relevés dans la partie 2.3, notamment au niveau des frontières des zones pluvieuses ou des cartes de pluies prédites à assez long terme qui sont parfois peut réalistes par rapport aux données en entrée (figure 2.6).

Annexe

Architecture détaillée du CNN_2D

Name	Kernel	Stride	Padding	Ch I/O	In Res	Out Res	Type	Input
conv1	4×4	2×2	1×1	12/16	128×128	64×64	Conv	in
conv2	4×4	2×2	1×1	16/64	64×64	32×32	Conv	conv1
conv3	4×4	2×2	1×1	64/128	32×32	16×16	Conv	conv2
conv4	4×4	2×2	1×1	128/192	16×16	8×8	Conv	conv3
conv5	4×4	2×2	1×1	192/256	8×8	4×4	Conv	conv4
deconv1	1×1	1×1	0×0	256/256	4×4	4×4	ConvTransp	conv5
deconv2	4×4	2×2	1×1	256/256	4×4	8×8	ConvTransp	deconv1
deconv3	4×4	2×2	1×1	256/128	8×8	16×16	ConvTransp	deconv2
deconv4	4×4	2×2	1×1	128/64	16×16	32×32	ConvTransp	deconv3
deconv5	4×4	2×2	1×1	64/24	32×32	64×64	ConvTransp	deconv4
deconv6	4×4	2×2	1×1	24/12	64×64	128×128	ConvTransp	deconv5

TABLE 2.3 – Architecture détaillée du CNN_2D.

Architecture détaillée du ConvGRU

Name	In Ker	Stride	In Padding	State Ker	State Dila	Ch I/O	In Res	Out Res	Type	Input	In State
conv1	4×4	2×2	1×1	1(3)/8	-	-	128×128	64×64	Conv	in	-
conv_gru1	3×3	1×1	1×1	8/64	3×3	1×1	64×64	64×64	ConvGRU	conv1	-
conv2	4×4	2×2	1×1	64/192	-	-	64×64	32×32	Conv	conv_gru1	-
conv_gru2	3×3	1×1	1×1	192/192	3×3	1×1	32×32	32×32	ConvGRU	conv2	-
conv3	4×4	2×2	1×1	192/192	-	-	32×32	16×16	Conv	conv_gru2	-
conv_gru3	3×3	1×1	1×1	192/192	3×3	1×1	16×16	16×16	ConvGRU	conv3	-
conv_gru4	3×3	1×1	1×1	192/192	3×3	1×1	16×16	16×16	ConvGRU	-	conv_gru3
deconv1	4×4	2×2	1×1	192/192	-	-	16×16	32×32	ConvTransp	conv_gru4	-
conv_gru5	3×3	1×1	1×1	192/192	3×3	1×1	32×32	32×32	ConvGRU	deconv1	conv_gru2
deconv2	4×4	2×2	1×1	192/64	-	-	32×32	64×64	ConvTransp	conv_gru5	-
conv_gru6	3×3	1×1	1×1	64/64	3×3	1×1	64×64	64×64	ConvGRU	deconv1	conv_gru1
deconv3	4×4	2×2	1×1	64/8	-	-	64×64	128×128	ConvTransp	conv_gru6	-
conv4	1×1	1×1	0×0	8/1	-	-	128×128	128×128	Conv	deconv3	-

TABLE 2.4 – Architecture détaillée du réseau ConvGRU.

L'entrée de ce réseau de neurones peut comporter 1 ou 3 canaux d'entrée si les données utilisées ne sont que des données de pluie ou si les données de vent sont incorporées.

Architecture détaillée du TrajGRU

Name	In Ker	Stride	In Padding	L	Ch I/O	In Res	Out Res	Type	Input	In State
conv1	4×4	2×2	1×1	-	1(3)/8	128×128	64×64	Conv	in	-
traj_gru1	3×3	1×1	1×1	13	8/64	64×64	64×64	TrajGRU	conv1	-
conv2	4×4	2×2	1×1	-	64/192	64×64	32×32	Conv	traj_gru1	-
traj_gru2	3×3	1×1	1×1	13	192/192	32×32	32×32	TrajGRU	conv2	-
conv3	4×4	2×2	1×1	-	192/192	32×32	16×16	Conv	traj_gru2	-
conv_gru3	3×3	1×1	1×1	9	192/192	16×16	16×16	TrajGRU	conv3	-
traj_gru4	3×3	1×1	1×1	9	192/192	16×16	16×16	TrajGRU	-	traj_gru3
deconv1	4×4	2×2	1×1	-	192/192	16×16	32×32	ConvTransp	traj_gru4	-
traj_gru5	3×3	1×1	1×1	13	192/192	32×32	32×32	TrajGRU	deconv1	traj_gru2
deconv2	4×4	2×2	1×1	-	192/64	32×32	64×64	ConvTransp	traj_gru5	-
traj_gru6	3×3	1×1	1×1	13	64/64	64×64	64×64	TrajGRU	deconv1	traj_gru1
deconv3	4×4	2×2	1×1	-	64/8	64×64	128×128	ConvTransp	traj_gru6	-
conv4	1×1	1×1	0×0	-	8/1	128×128	128×128	Conv	deconv3	-

TABLE 2.5 – Architecture détaillée du réseau TrajGRU.

L'entrée de ce réseau de neurones peut comporter 1 ou 3 canaux d'entrée si les données utilisées ne sont que des données de pluie ou si les données de vent sont incorporées.

Références

[1] R. E. Rinehart et E. T. Garvey. "Three-dimensional storm motion detection by conventional weather radar". In : *Nature* 273 (1978), p. 287-289.

[2] L. Li, W. Schmid et J. Joss. "Nowcasting of Motion and Growth of Precipitation with Radar over a Complex Orography." In : *Journal of Applied Meteorology* 34.6 (juin 1995), p. 1286-1300.

[3] WANG Gaili et al. "Application of Multi-Scale Tracking Radar Echoes Scheme in Quantitative Precipitation Nowcasting". In : *ADVANCES IN ATMOSPHERIC SCIENCES* 30.120026 (2013), p. 448. ISSN : 0256-1530. DOI : 10.1007/s00376-012-2026-7. URL : http://www.iapjournals.ac.cn/aas//article/id/266.

[4] Hana Kyznarová et Petr Novák. "Development Of Cell-Tracking Algorithm In The Czech Hydrometeorological Institute". In : (jan. 2005).

[5] Wang-chun Woo et Wai-kin Wong. "Operational Application of Optical Flow Techniques to Radar-Based Rainfall Nowcasting". In : *Atmosphere* 8.3 (2017). ISSN : 2073-4433. DOI : 10.3390/atmos8030048. URL : https://www.mdpi.com/2073-4433/8/3/48.

[6] Marc'Aurelio Ranzato et al. "Video (language) modeling : a baseline for generative models of natural videos". In : *ArXiv* abs/1412.6604 (2014).

[7] Xingjian Shi et al. "Convolutional LSTM Network : A Machine Learning Approach for Precipitation Nowcasting". In : *NIPS*. 2015.

[8] Xingjian Shi et al. *Deep Learning for Precipitation Nowcasting : A Benchmark and A New Model*. 2017. arXiv : 1706.03458 [cs.CV].

[9] Lei Han et al. "Convective Precipitation Nowcasting Using U-Net Model". In : *IEEE Transactions on Geoscience and Remote Sensing* (2021), p. 1-8. DOI : 10.1109/TGRS.2021.3100847.

[10] Vincent Bouget et al. "Fusion of Rain Radar Images and Wind Forecasts in a Deep Learning Model Applied to Rain Nowcasting". In : *Remote Sensing* 13.2 (jan. 2021), p. 246. DOI : 10.3390/rs13020246. URL : https://hal.sorbonne-universite.fr/hal-03112093.

[11] Suman V. Ravuri et al. "Skilful precipitation nowcasting using deep generative models of radar". In : *Nature* 597 (2021), p. 672-677.

[12] G. Larvor, L. Berthomier, V. Chabot, B. Le Pape, B. Pradel, L. Perez. *METEO FRANCE Data Playground.* https://meteonet.umr-cnrm.fr/.

[13] Aniss Zebiri. "Estimation de la dynamique atmospherique à partir de l'assimilation d'images radars multi-echelles. Application à la prevision des taux de precipitation à courte echeance, à partir des images radars." Theses. Sorbonne Université, juin 2020. URL : https://hal.sorbonne-universite.fr/tel-03127158.