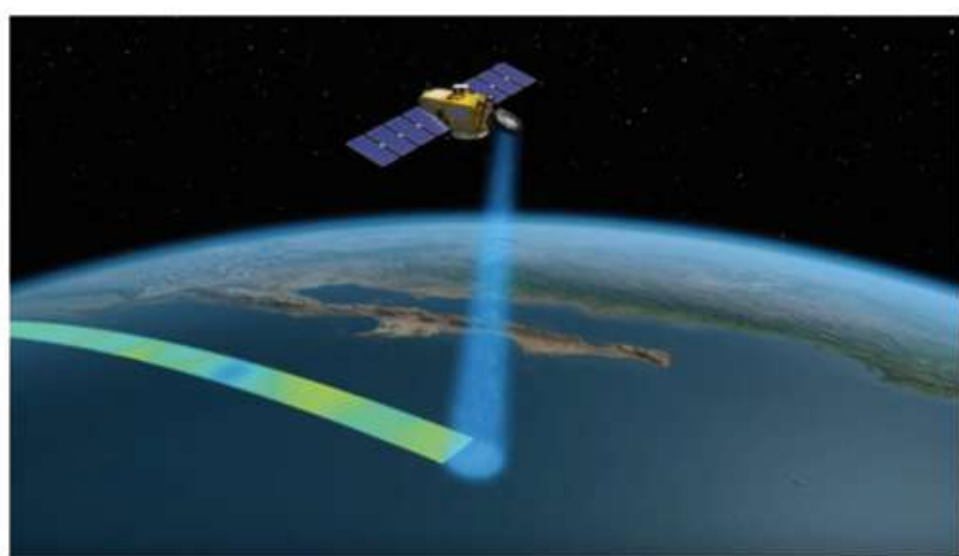


# DESCENTE D'ÉCHELLE SUR L'OCÉAN PAR FUSION D'OBSERVATIONS HÉTÉROGÈNE EN UTILISANT DES MÉTHODES D'APPRENTISSAGE PROFOND

Encadrement : A. Charantonis (LOCEAN), C. Sorrow, C. Mejia (ESPRI\_IA), S. Thiria (LOCEAN)

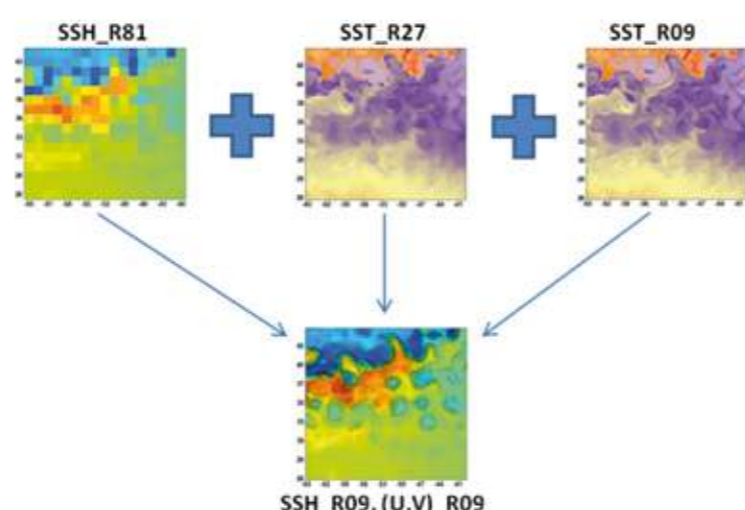
## I - Problématique



Les mesures réalisées par satellites permettent d'obtenir la SST (Sea Surface Temperature) avec une bonne résolution (5km \* 5km) au contraire de la SSH (Sea Surface Height) observée à une résolution plus basse (25km \* 25km) dont la haute résolution est difficilement accessible avec les appareils de mesures actuels.

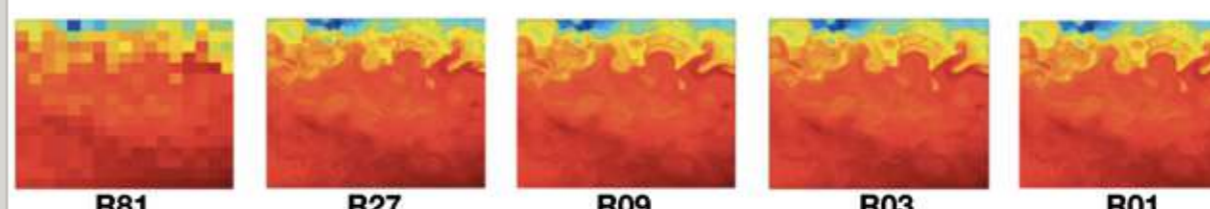
Le but de ces travaux de recherche est d'augmenter artificiellement et graduellement la résolution de SSH en s'appuyant sur les données de SST connues aux résolutions intermédiaires.

En plus de la SSH, le système devra restituer les composantes  $\vec{U}$  (Est-Ouest) et  $\vec{V}$  (Nord-Sud) du courant marin qui se déduisent de la combinaison de SSH et de SST.



Schématisme des attentes du fonctionnement de l'algorithme

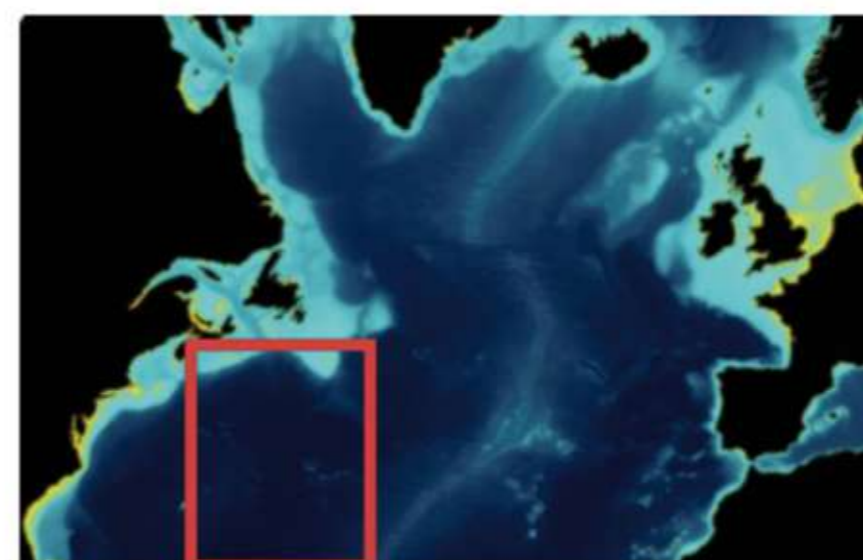
## II - Développement de la méthode de descente d'échelle



Différentes résolutions de SST du modèle NATL60

Voici les différentes résolutions que nous avons définies.

Résolution	Nombre de pixels	Résolution en kilomètres <sup>2</sup>
R01	1296*1377	1.5*1.5
R03	432*459	4.5*4.5
R09	144*153	13*14
R27	48*51	40*41
R81	16*17	120*122



Zone d'étude encadrée : Une partie de la région du Gulf Stream (26°N,45°N ; 40°W,65°W)

Pour ce faire, nous utilisons des données simulées par le modèle NATL60. Ce dernier caractérise les phénomènes complexes qui interviennent dans les océans. Ces données océanographiques journalières sont simulées du 1er Octobre 2012 jusqu'au 1er Octobre 2013. Nous disposons donc de la SSH, de  $\vec{U}$  et de  $\vec{V}$  aux 5 résolutions ci dessus.

Pour développer cette méthode, nous allons utiliser des algorithmes d'apprentissage profond dit de "Super Résolution".

## III - L'algorithme de SuperResolution : ResacNet

Pour faire fonctionner notre réseau de neurones, nous divisons nos données (366 images) en 3 catégories :

- 256 images par variable pour l'apprentissage (70%)
- 55 images par variable pour la validation (15%)
- 55 images par variable pour le test (15%)

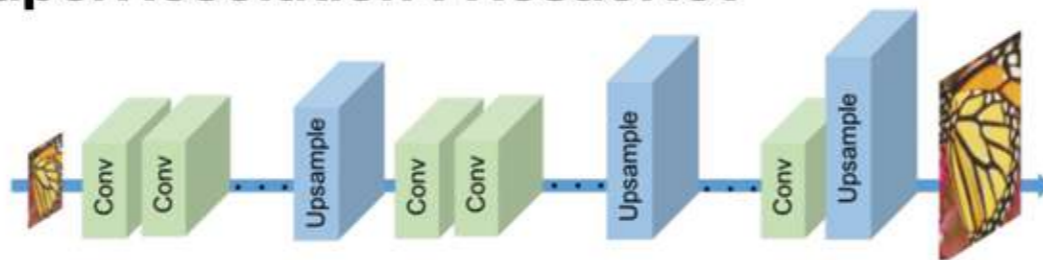
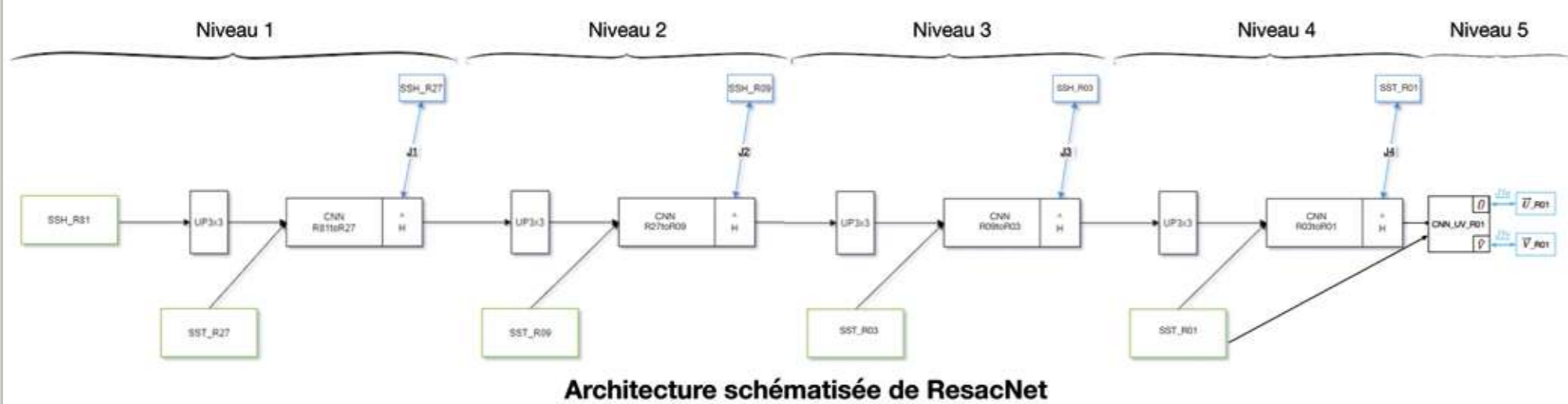


Schéma de l'architecture : Progressive Upsampling SuperResolution

L'architecture simplifiée du réseau de neurones est présentée dans ces figures. Ce réseau est assez particulier car en plus de la SSH en entrée à la résolution inférieure, il faut intégrer la SST à la résolution supérieure. De plus, chaque stage du réseau à sa propre fonction de coût  $J_i$ .

On retrouve une structure répétitive avec des CNN (Convolutional Neural Networks) à la suite, entremêlée d'Upsampling (UP) pour améliorer progressivement la résolution de notre image. À chaque étape, la résolution de l'image est augmentée de (3x3).



Architecture schématisée de ResacNet

## IV - Paramétrisation du modèle Resac

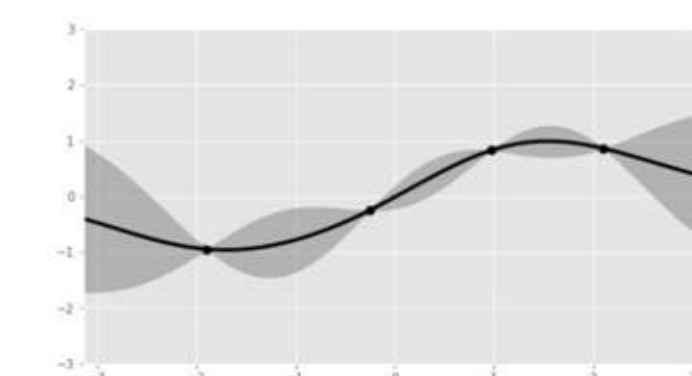
Un réseau de neurones possède de nombreux paramètres qu'il faut fixer. La définition optimale de ces paramètres permet d'améliorer les performances des résultats (erreurs minimales entre les valeurs prédites et réelles) et/ou celles des temps/puissances de calcul (ces réseaux de neurones possèdent des milliers de poids et leur entraînement des durées de calculs importantes).

À chaque itération, une combinaison d'hyperparamètres est choisie et un nouveau point est associé à la fonction de coût. À partir de cela, des fonctions de substitution émergent à l'aide de processus gaussiens estimant différentes allures de la fonction.

Ces combinaisons de paramètres présentent une difficulté car elles sont infinies et il faut passer par une méthode d'optimisation pertinente pour trouver le bon agencement des paramètres.

### Optimisation bayésienne

L'optimisation bayésienne est la méthode qui a été utilisée. Elle permet d'approcher les minimums globaux des fonctions de coûts du réseau en très peu d'étapes de calculs. Elle repose sur l'approximation de la fonction de coût par une fonction de substitution basée sur différents points échantillonnés.



Fonction objective et les intervalles de confiance estimés

Au final, on obtient une estimation de la fonction de coût, avec un certain intervalle de confiance, en fonction des différentes combinaisons d'hyperparamètres du réseau. Ceci permet d'en extraire le minimum et la combinaison optimale.

Par ailleurs, cette méthode nécessite un équilibre entre exploration (prendre des paramètres variés pour explorer tout le spectre disponible) et exploitation (se concentrer sur les régions où la fonction de coût est basse).

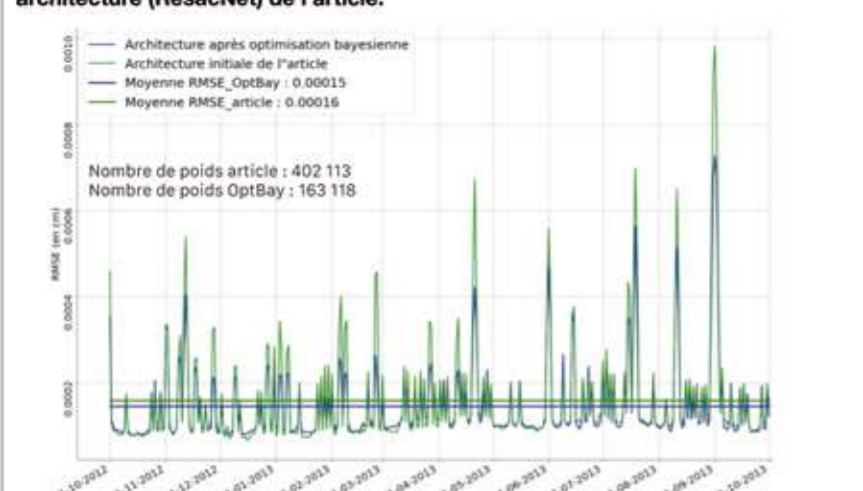
Exemple de différentes fonctions de substitutions

## V - Résultats

Pour l'optimisation bayésienne, les hyperparamètres que nous avons choisis sont le nombre de convolutions, le nombre de filtres par couche de convolution, le learning rate, la fonction de sortie du réseau.

Tous les calculs menés et présentés dans cette section ont été réalisés jusqu'à la résolution R09 par gain de temps (il a été vérifié que le réseau fonctionne jusqu'à R01).

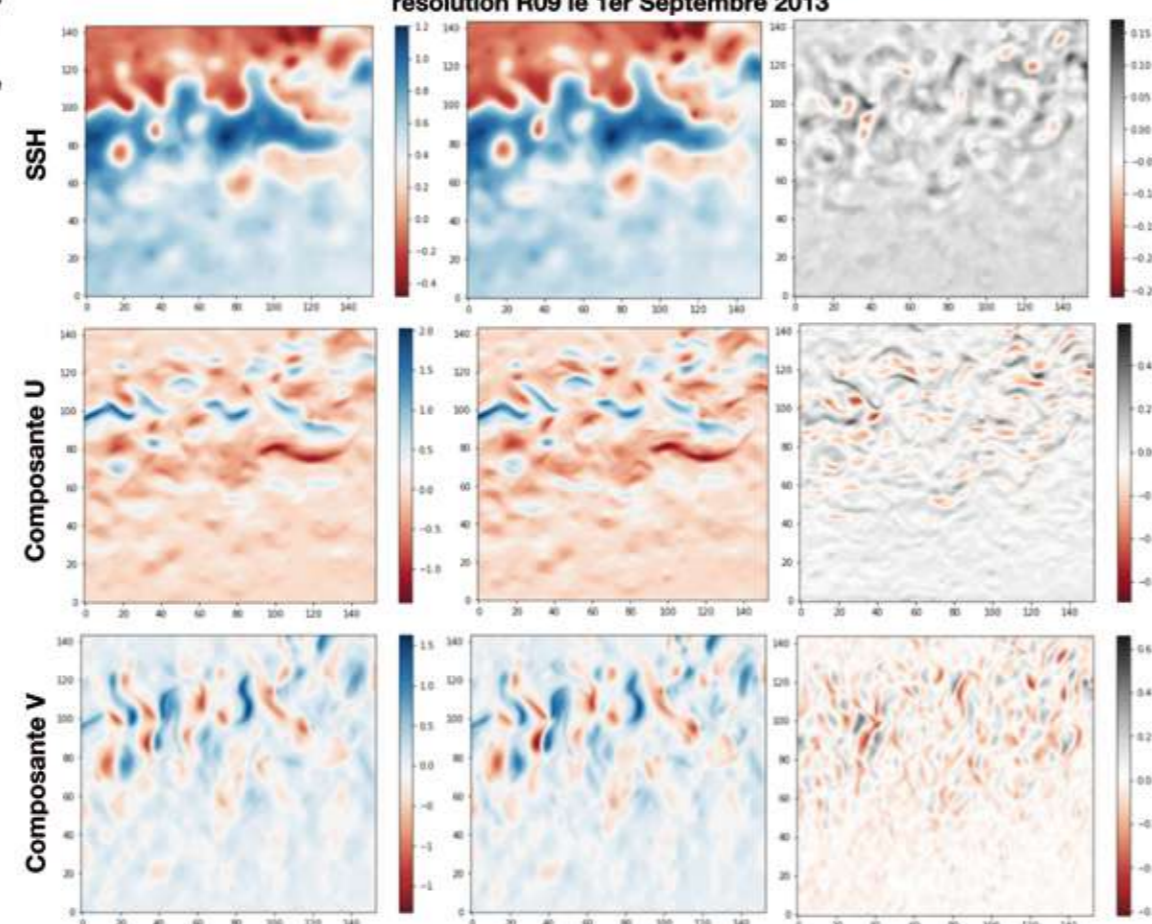
Une fois l'optimisation bayésienne réalisée, nous avons une nouvelle architecture (ResacBayes) dont nous pouvons comparer les performances avec l'ancienne architecture (ResacNet) de l'article.



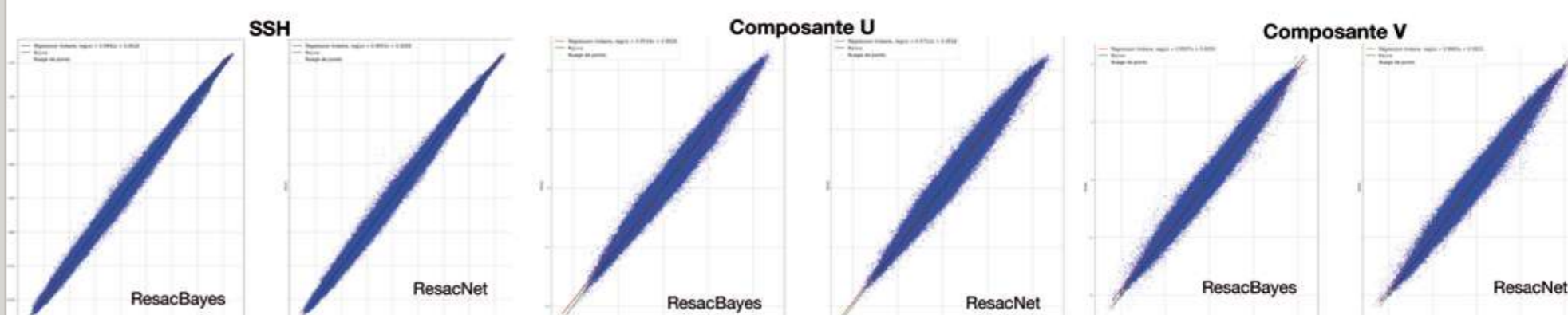
RMSE journalière de prédiction de la SSH à R09 du 1er Octobre 2012 au 1er Octobre 2013 pour les deux architectures

La RMSE journalière sur des prédictions de SSH permet de rendre compte des performances du réseau de neurones. On remarque ici une légère amélioration de l'erreur moyenne. Cependant les points les plus marquants sont le nombre de poids requis par l'architecture (réduit de 60%) ainsi que la diminution de la sensibilité aux "larges erreurs" (les pics) qui peuvent s'expliquer par l'ajout de couche de Batch Normalisation pour rendre plus "stable" les prédictions.

Prédiction, modèle et différence modèle/prédiction (de gauche à droite), de la SSH (en m), de la composante de U (en  $m \cdot s^{-1}$ ) et de la composante de V (en  $m \cdot s^{-1}$ ) à la résolution R09 le 1er Septembre 2013



## VI - Performances



Chaque point sur ces graphiques représente un pixel avec en abscisse sa valeur sur les données modèles et en ordonnées sa valeur prédite par l'algorithme. À gauche, on retrouve l'architecture ResacBayes et à droite ResacNet.

En réalisant des régressions linéaires sur ces nuages de points, on obtient des coefficients de détermination R2 supérieur à 96% allant même jusqu'à plus de 99% pour la SSH.

Ces graphiques permettent de démontrer l'efficacité du réseau de neurones mis en place et la qualité de ses prédictions.

Finalement, l'optimisation bayésienne a permis d'améliorer les performances du réseau en diminuant drastiquement le nombre de poids (60%) et en rendant les prédictions plus stables.

Les perspectives suivantes de ces travaux sont la réalisation de l'optimisation bayésienne pour les autres niveaux de l'architecture (R09-R03 et R03-R01) ainsi que des essais de l'architecture sur des données bruitées pour au final utiliser ResacBayes sur des données satellite réelles.

Références bibliographiques : Andre YE, *The Beauty of Bayesian Optimization, Explained in Simple Terms (2020)*  
Zhihao Wang, Jian Chen, Steven C.H. Hoi, Fellow, IEEE, *Deep Learning for Image Super-resolution: A Survey (2020)*