



# Compte rendu 7

*BELDJILALI Maxime, CHATEAUNEUF Arthur*

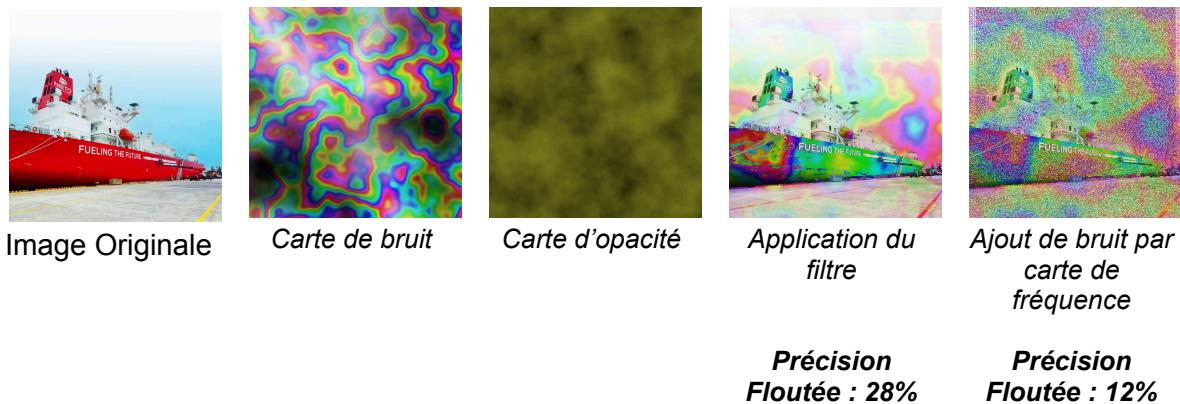
<b>Attaque des basses fréquences</b>	<b>2</b>
Méthode de Perlin	2
Amélioration et préservation légère de la teinte	2
<b>Conclusion de tous nos filtres</b>	<b>3</b>
Liste complète	3
Remarques et Conclusion	4
<b>Annexes</b>	<b>5</b>
Dépot Github	5

# Attaque des basses fréquences

## Méthode de Perlin

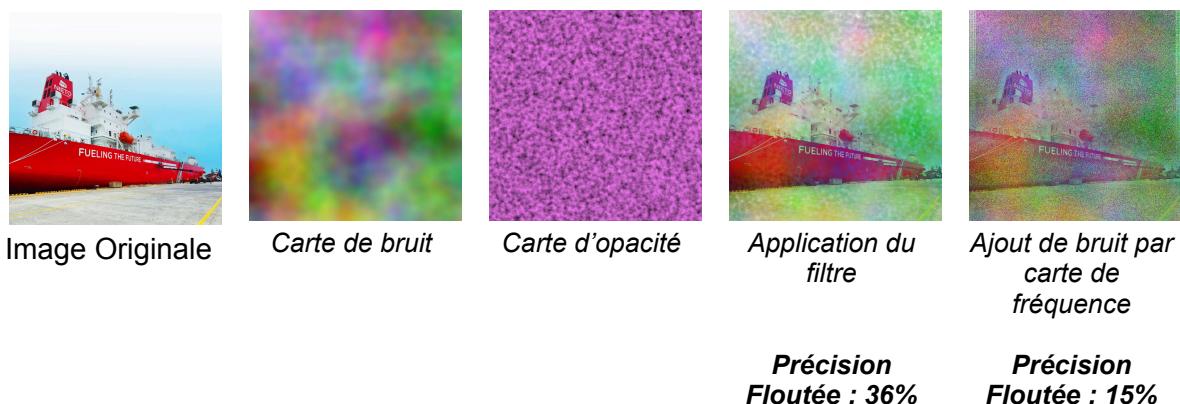
Nous souhaitons, à partir de nos meilleurs résultats, créer un filtre tout aussi efficace contre les basses fréquences que les hautes fréquences. C'est-à-dire que l'efficacité de notre CNN soit proche de l'aléatoire avec ou sans flou de prétraitement.

Pour cela, nous introduisons, en plus du bruit par carte de fréquence, un bruit moyen et basse fréquence grâce à au bruit de Perlin. Notre premier filtre utilisant cette méthode exploite l'espace couleur HSL. Nous commençons par créer une carte de bruit HS ainsi qu'une carte d'opacité, utilisant un bruit de perlin à une échelle différente.



## Amélioration et préservation légère de la teinte

Nous souhaitons améliorer ce filtre en préservant un maximum d'information de teinte, tout en gardant une reconnaissance du CNN minimum. Pour cela, nous appliquons une méthode similaire dans l'espace RGB, en donnant une opacité moins élevée du bruit sur le vert.



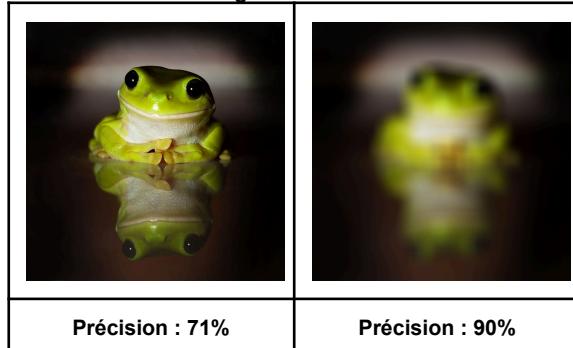
Nous obtenons un filtre aux résultats similaires, mais préservant mieux les informations de l'image pour un observateur humain.

# Conclusion de tous nos filtres

## Liste complète

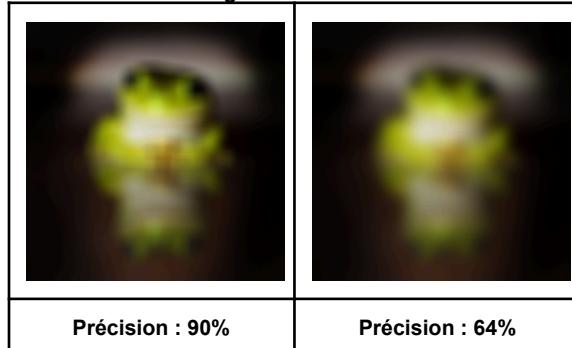
Voici la liste de nos 9 filtres, avec leur précision avec et sans prétraitement de flou :

Fig1 : Sans filtre



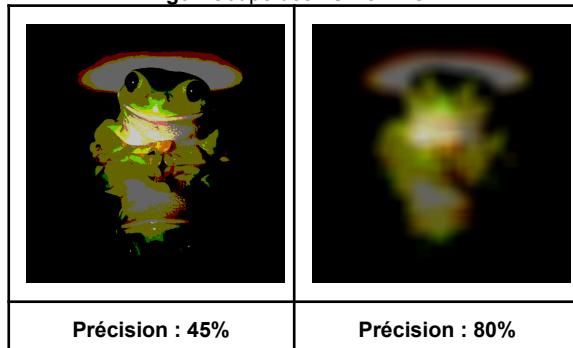
Note : Un léger flou extrait mieux les basses fréquences et permet une meilleure reconnaissance du CNN.

Fig2 : Flou intense



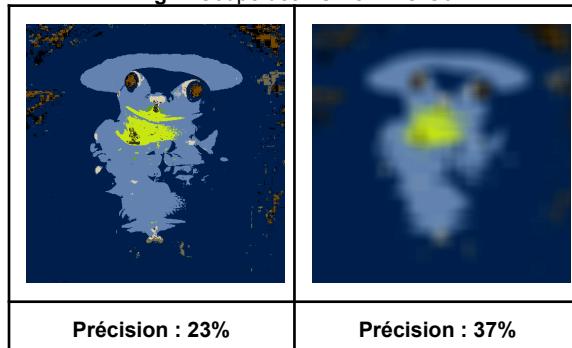
Note : Un flou trop intense détériore la reconnaissance.

Fig3 : Coupe des LSB en RGB



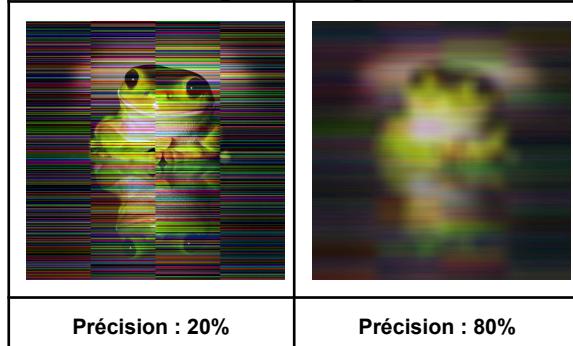
Note : Faible à l'analyse par histogramme.

Fig4 : Coupe des LSB en YrCrCb



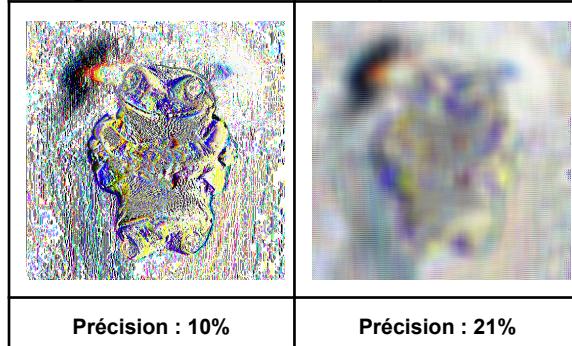
Note : Faible à l'analyse par histogramme. Attaquer la teinte permet de meilleurs résultats.

Fig5 : Bruit en ligne



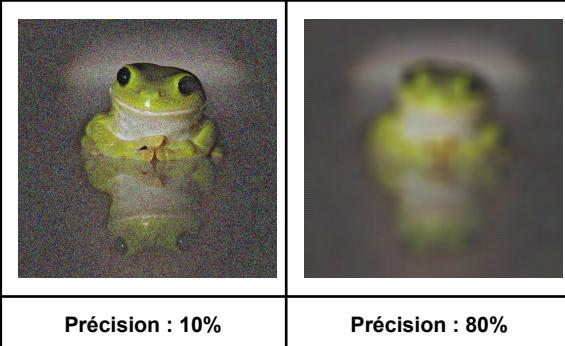
Note : Très efficace uniquement lorsque le CNN utilise les basses fréquences.

Fig6 : Division des contours en espace ondulatoire



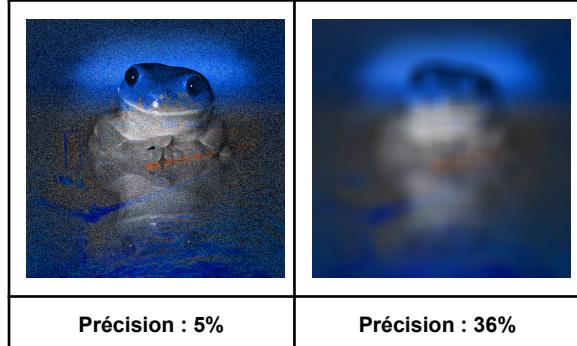
Note : Faible à l'analyse par histogramme. Très efficace mais trop de dégradation pour un observateur humain.

**Fig7 : Bruitage par carte de fréquence**



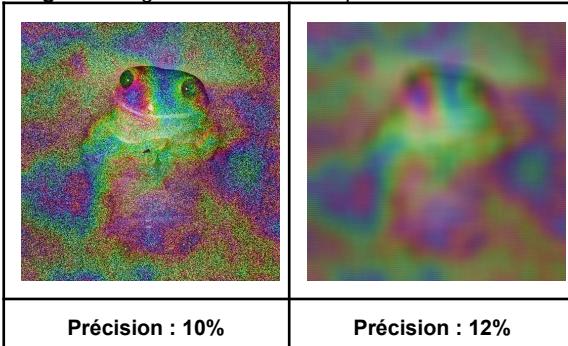
**Note :** Conserve bien la qualité de l'image. Efficace uniquement pour les CNN basses fréquences.

**Fig8 : Empoisonnement de la teinte et saturation**



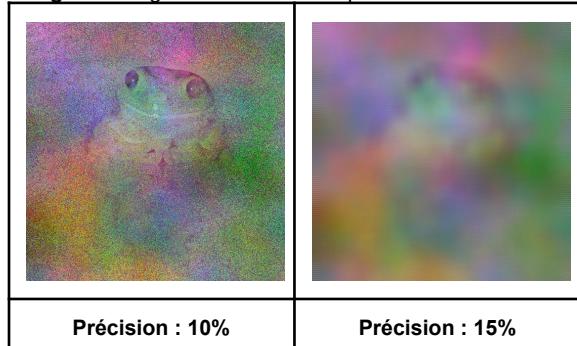
**Note :** Meilleure efficacité sur les basses fréquences avec une précision du CNN jusqu'à plus de deux fois inférieur à l'aléatoire.

**Fig8 : Bruitage de la teinte et sat. par méthode de Perlin**



**Note :** Faible à l'analyse par histogramme. Efficace pour toutes les plages de fréquence.

**Fig2 : Bruitage non linéaire RGB par méthode de Perlin**



**Note :** Faible à l'analyse par histogramme. Efficace pour toutes les plages de fréquence. Conserve bien mieux les informations de teinte pour un observateur humain que le filtre précédent.

## Remarques et Conclusion

Nous avons obtenu une série de filtres efficaces pour plusieurs utilisations, qui ont chacuns le même but d'obscuration. Durant la suite de ce projet, nous souhaitons évaluer des observateurs humains sur ces derniers.

Nous prévoyons ainsi de créer un formulaire en ligne, permettant aux participants de répondre à des questions de reconnaissance du sujet, ou de détails de l'image, sous plusieurs filtres. Nous voulons demander aux participants de noter leur niveau de confiance sur les réponses qu'ils donneront.

Nous avons également pour objectif de livrer une application ayant une interface utilisateur simple permettant d'appliquer les filtres sur n'importe quelle image, en donnant les informations adaptées à l'efficacité potentielle de chaque filtre dépendant la nature du CNN contre lequel l'utilisateur veut se protéger.

# **Annexes**

## Dépot Github

- <https://github.com/Pataleon/Securite-visuelle>