

Compte rendu de deep learning TP 9

**Elodie Difonzo
Roqyun Ko**

Partie 1 Carte de saillance

1. **Montrer et interpréter les résultats obtenus**
2. **Discutez les limites de cette technique pour visualiser l'importance des différents pixels**


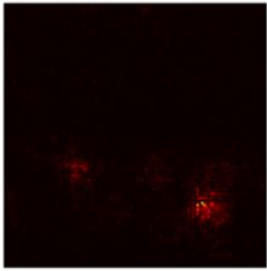
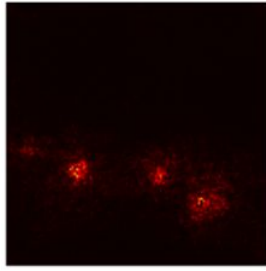

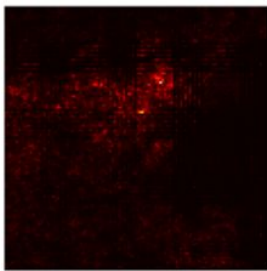
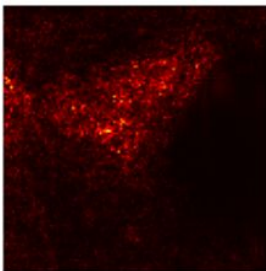
On observe que lorsqu'il y a peu de contraste entre les différents objets de l'image cette technique ne permet pas de les distinguer correctement (ils semblent parfois confondus).

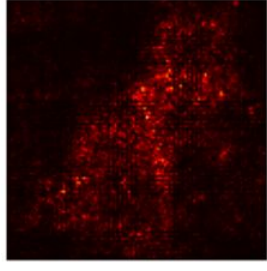
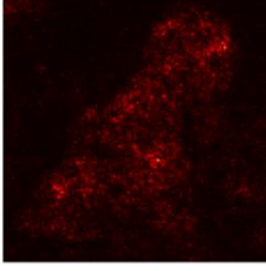

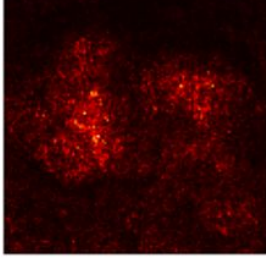
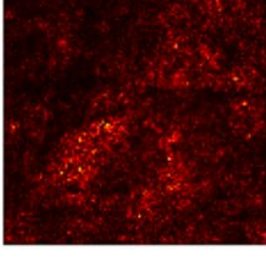
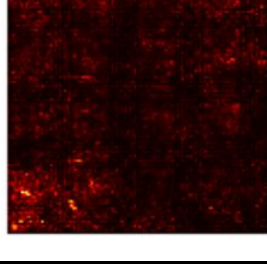
Le réseau de neurones est non linéaire alors que la fonction utilisée pour produire la carte de saillance est linéaire. C'est pour cette raison que la carte de saillance ne peut pas représenter correctement le comportement du réseau.

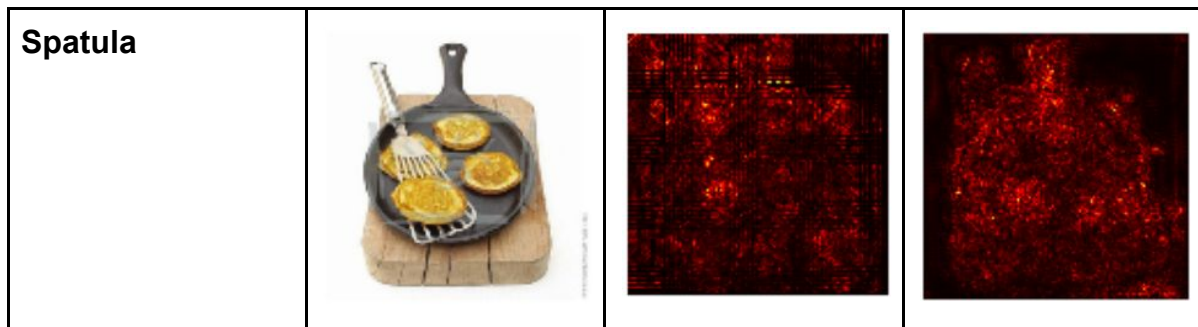
3. **Cette technique pourrait-elle servir à autre chose qu'à interpréter le réseau ?**

Cette technique peut être utilisée pour la localisation d'objets dans une image.

4. **Bonus :Tester avec un autre réseau, par exemple VGG16.**

	Original	SqueezeNet	VGG16
Hay			
Quail			

Tibetan mastiff			
Border terrier			
Brown bear			
Pyjama			
Sports car			



Partie 2 Exemples adversaires

5. Montrer et interpréter les résultats obtenus.

Iterations : 8
 True class : 85 / quail
 Fooled class : 6 / stingray



La génération d'une fooling image avec 8 itérations est suffisante pour duper le réseau. En revanche, la différence entre l'image originelle et l'image dupée n'est pas du tout importante. Elle n'est visible qu'après avoir amplifié la différence.

En attaquant le réseau avec les régions activées d'une fausse cible, on peut facilement écraser les features importants de la vraie cible. Ainsi, on peut mener le réseau à mal classer les images.

6. Quelles conséquences cela peut-il avoir pour l'utilisation de réseaux de convolution en pratique ?

Aujourd'hui, on utilise les réseaux de neurones pour des systèmes autonomes comme des voitures autonomes. Par exemple, si jamais un attaquant réussit à duper un système autonome d'une voiture, il peut bien causer un accident.

On pourra utiliser cette technique pour augmenter les données.

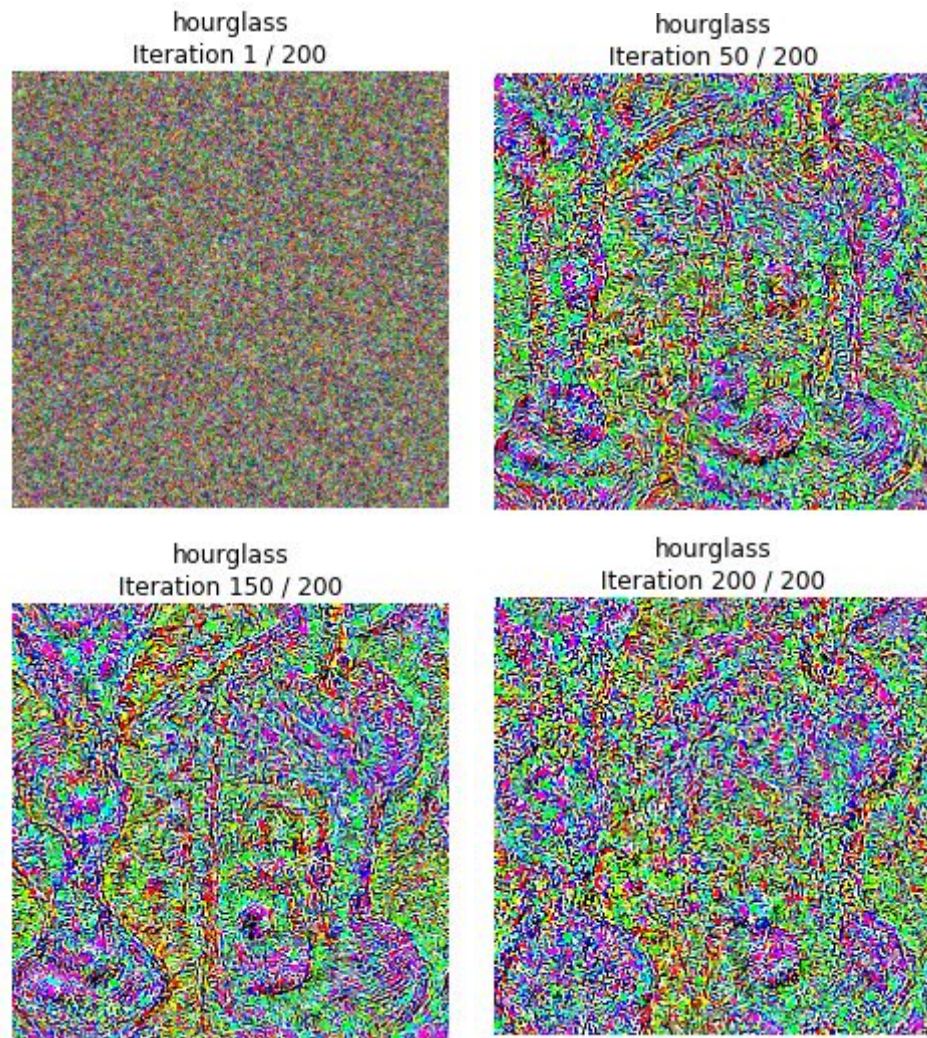
7. Bonus : Discutez des limites de cette façon naïve de construire des images adversaires. Proposez éventuellement des alternatives / extensions. (Vous pouvez vous inspirer de la littérature)

La limite de cette technique est qu'on peut entraîner le réseau pour ne pas se tromper en ajoutant les "fooled images" dans l'ensemble de train. Il faut aussi le modèle du réseau pour ce genre d'entraînement car l'attaque adverse est faite à partir des études de comportements du réseau.

Mais il existe encore de nombreuses manières pour duper les réseaux de convolution. Par exemple, on peut tout simplement attaquer les images avec des bruits uniformes, dessiner un rectangle blanc, etc.

8. Montrer et interpréter les résultats.

$$\lambda = 1e-3, \eta = 5$$



On commence par une image avec des pixels aléatoires. Après la création de l'image, on "jitter" cette image et on exécute la forward et la backward propagation avec l'image. Puis, on ajoute le gradient régularisé avec la norme L2 de l'image à la même image. On brouille l'image. On répète cette procédure pour obtenir une visualisation d'une classe. On observe que la morphologie en huit devient de plus en plus claire dans les figures ci-dessus. Après chaque itération, les neurones (pixels) deviennent de plus en plus activés pour que l'image corresponde à la classe cible.

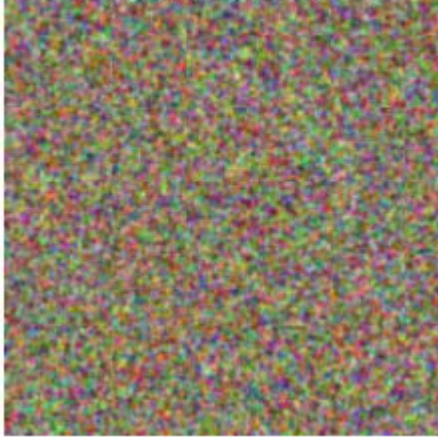


Une démonstration du filtre “jitter”. (Source :
<https://demonstrations.wolfram.com/ImageJitterFilter/>)

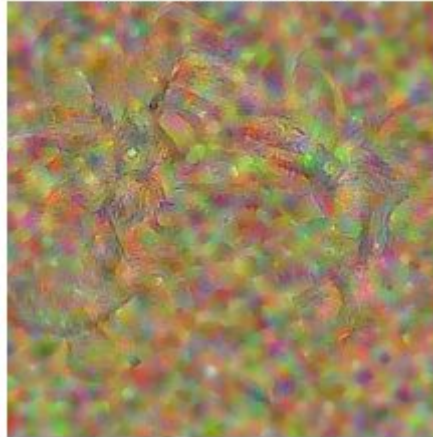
9. Essayer de varier le nombre d'itérations et le learning rate, le poids de la régularisation.

$$\lambda = 1e - 4, \eta = 4e - 3$$

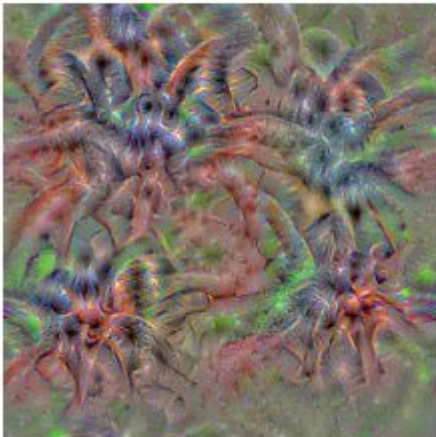
tarantula
Iteration 50 / 5000



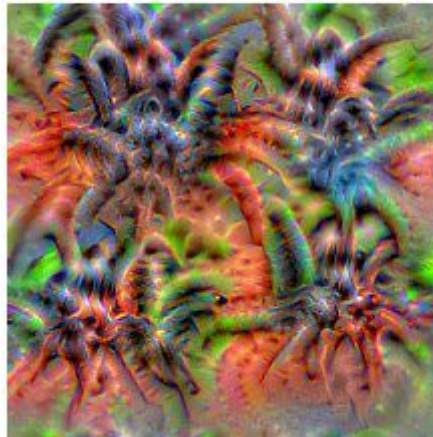
tarantula
Iteration 300 / 5000



tarantula
Iteration 1000 / 5000



tarantula
Iteration 5000 / 5000



$$\lambda = 5e - 5, \eta = 4e - 3$$

hourglass
Iteration 5000 / 5000



Yorkshire terrier
Iteration 2500 / 5000



10. Essayer d'utiliser une image d'ImageNet comme image source au lieu d'une image aléatoire (paramètre init_img). Vous pouvez utiliser pour classe cible la classe réelle. Commenter l'intérêt.

$$\lambda = 5e - 5, \eta = 4e - 3$$

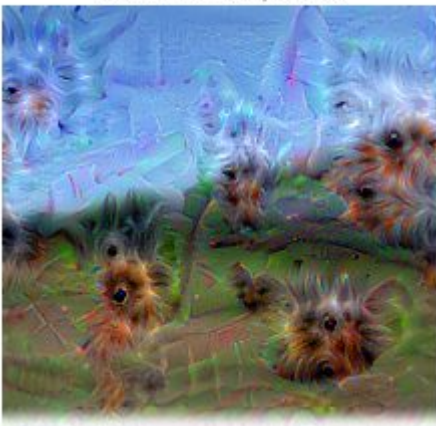
Yorkshire terrier
Iteration 1 / 2500



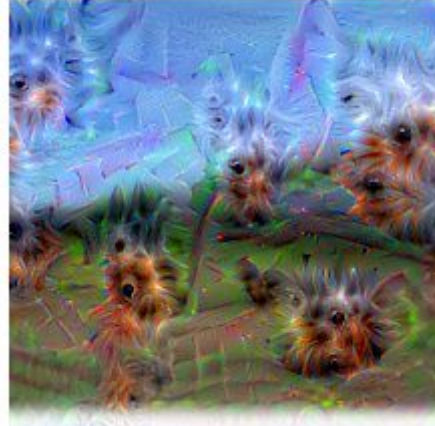
Yorkshire terrier
Iteration 200 / 2500



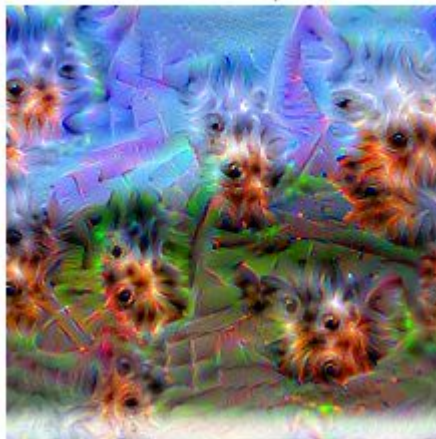
Yorkshire terrier
Iteration 500 / 2500



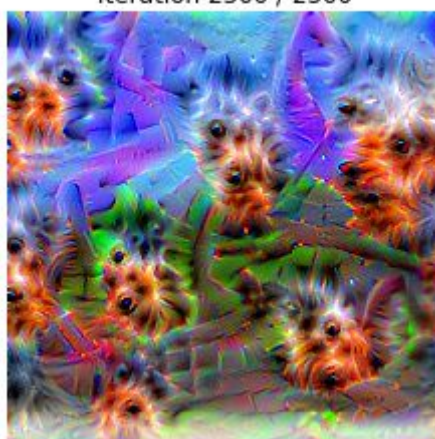
Yorkshire terrier
Iteration 650 / 2500



Yorkshire terrier
Iteration 1250 / 2500



Yorkshire terrier
Iteration 2500 / 2500



$$\lambda = 5e - 5, \eta = 4e - 3$$

hay
Iteration 1 / 2500



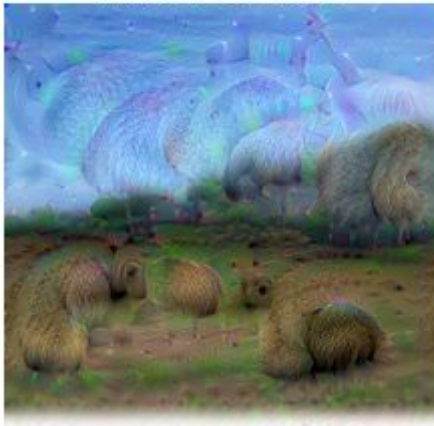
hay
Iteration 125 / 2500



hay
Iteration 300 / 2500



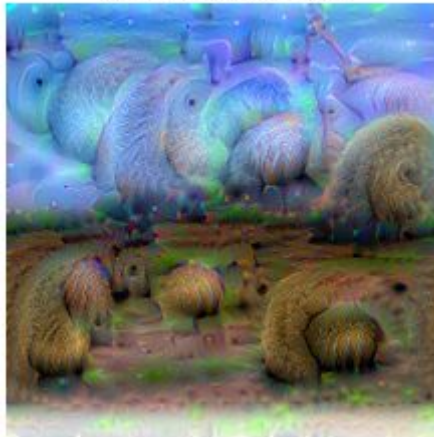
hay
Iteration 650 / 2500



hay
Iteration 1250 / 2500



hay
Iteration 2500 / 2500



L'intérêt de la visualisation de classes est principalement la génération d'une image. En construisant un réseau de discriminateur qui vérifie la qualité d'image, on pourra mieux générer les images vraies.

11. Bonus : Essayer avec un autre réseau, par exemple VGG16.

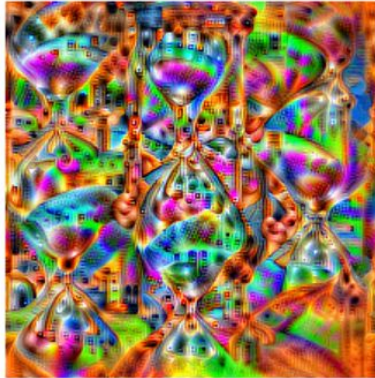
$$\lambda = 5e - 5, \eta = 4e - 3$$

$$\lambda = 1e - 3, \eta = 9e - 4$$

hourglass
Iteration 5000 / 5000



hourglass
Iteration 8225 / 10000



$$\lambda = 1e - 3, \eta = 9e - 4$$

hourglass
Iteration 1075 / 100000



hourglass
Iteration 1500 / 100000



hourglass
Iteration 3525 / 100000



hourglass
Iteration 8300 / 100000

