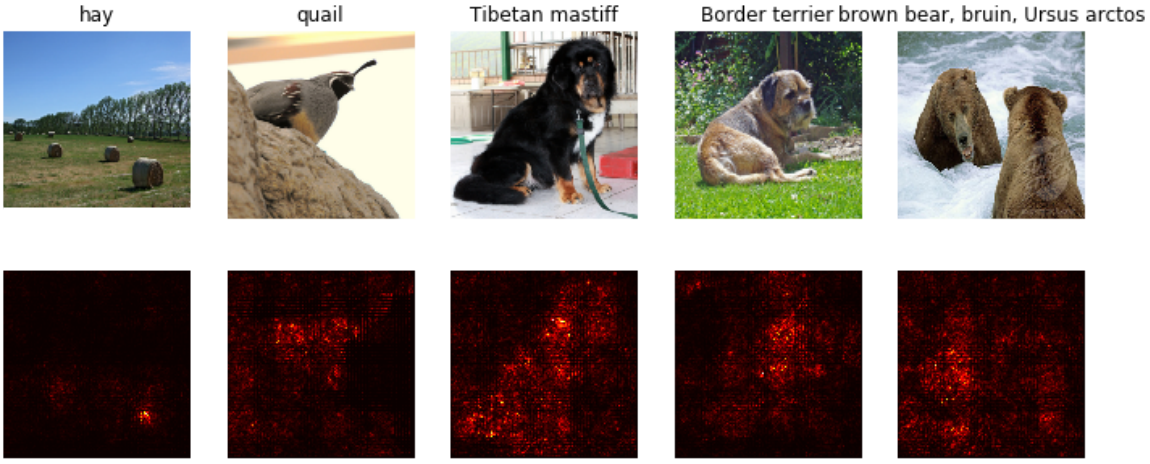
Keyvan Beroukhim
Arij Riabi

TP 9 - Visualisation des réseaux de neurones

1) **Les cartes de saillance sont activées au niveau d'éléments caractéristiques** des objets de l'image. Par exemple au niveau de la tête des chien ou des bottes de foin.

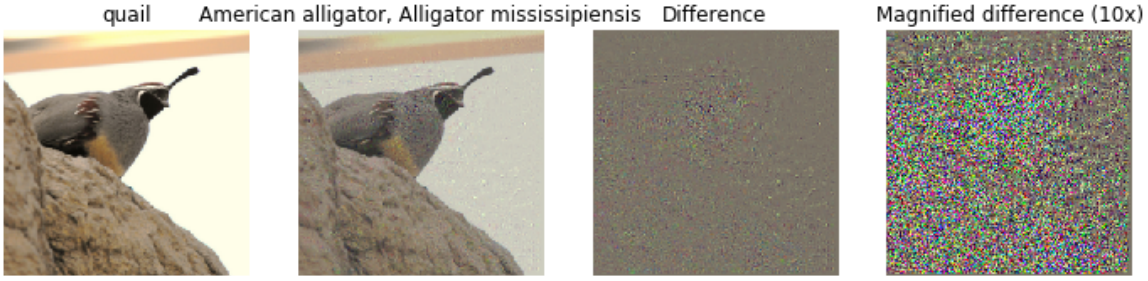


2) Les activations de la carte de saillance **ne recouvrent pas totalement les zones** qu'un humain caractérisait comme importantes (ex: les autres bottes de foin). Cette technique souffre d'une **forte variance**: en changeant l'image, les activations peuvent changer fortement. Finalement, cette technique suppose la linéarité du réseau.

3) Cette technique peut servir à la **détection d'objets** ou à la **segmentation d'image**.

4) En utilisant un autre réseau de neurone convolutif, on visualisera forcément des images du même type car les convolutions auront la même expressivité.

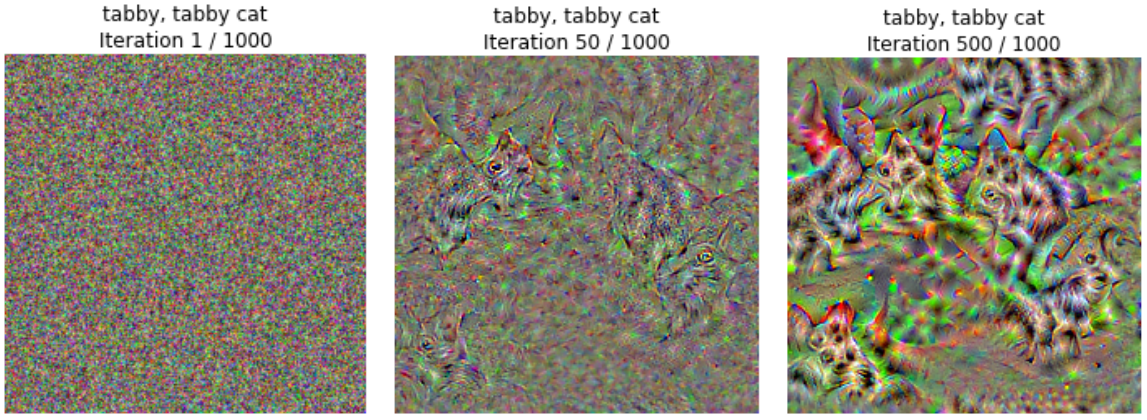
5) Quelle que soit l'image en entrée et quelle que soit la classe choisie, **on arrive à tromper le réseau de neurone sans qu'un utilisateur humain ne voit de réelle différence**. Par exemple on a réussi à construire une image d'oiseau que le réseau de neurone interprète comme un alligator. Les changements effectués à l'image ne sont pas naturels, nous créons une nouvelle dynamique d'image en rajoutant un bruit que le réseau n'est pas habitué à voir.



6) Nous venons de démontrer que l'utilisation de ces réseaux de neurones n'est pas fiable si nous avons accès au modèle (et au dataset). On peut par exemple penser à des algorithmes de détection automatique de panneaux de signalisation pour voiture autonome. En appliquant une modification difficilement perceptible pour les humains sur le panneau, on pourrait tromper les voitures.

7) En pratique le modèle n'est pas souvent accessible. Ajouter un motif ayant des features très caractéristiques d'une classe permet d'avoir de très bonnes chances de tromper l'algorithme.

8) De la manière que précédemment, on modifie l'image en suivant le gradient. On ne cherche plus à tromper le réseau mais juste à créer une image appartenant à une classe. En partant d'un bruit, cette technique fait apparaître des traits caractéristiques de la classe choisie. Dans l'exemple ci-dessous, des pelages de chat et des formes d'oreilles de chat apparaissent.



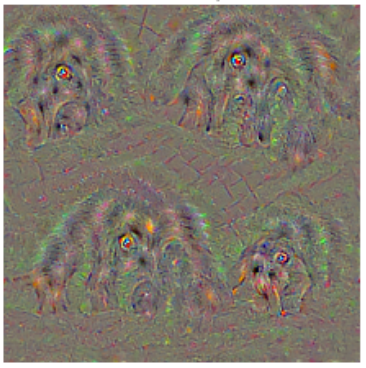
9-10) Afin d'obtenir des images plus réalistes, on remplace le bruit initial par une image représentant un objet de la classe à illustrer. La technique est alors supposée amplifier les traits caractéristiques déjà présents dans l'image.

image en entrée

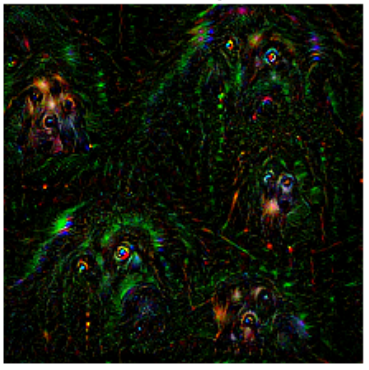
après quelques epoch, certains traits caractéristiques sont accentués, des artefacts apparaissent déjà

sans régularisation

régularisation L2
obtenue par une de la distance à l'image initiale
erreur d'implémentation



régularisation L2
de la norme de la valeur des pixels



Image

Sans régularisation sur la norme de la valeur des pixels, les images deviennent psychédéliques, si la régularisation est trop forte, les images deviennent ternes. Il n'y a pas d'entre deux qui permette de générer des images réalistes.
Augmenter le nombre d'epoch et faire baisser progressivement le learning rate n'a pas d'impact notable, l'image obtenue oscille autour d'optimums locaux.
Utiliser une norme L2 de la distance par rapport à l'image initiale pourrait permettre d'amplifier les traits sans trop s'éloigner de l'image originale. Mais pour un oeil humain, l'image obtenue en pratique **ressemble moins à la classe cible que l'image originale**.