Projet Image - Compte-rendu 4

Sécurité Visuelle - Obscuration d'image

Loïc Kerbaul - Valentin Noyé

11 novembre 2024

1 Introduction

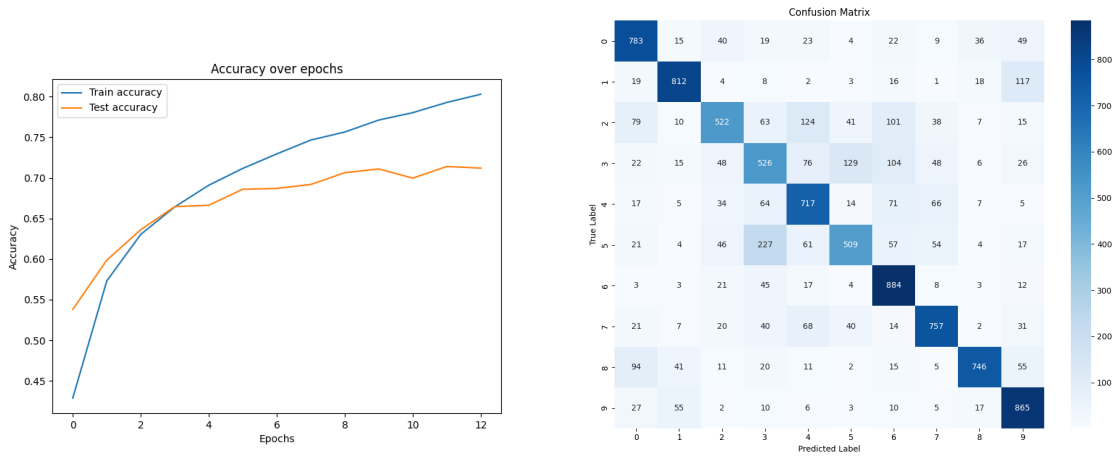
Ce quatrième compte-rendu se consacre à l'évaluation de l'offuscation par réseau de neurones et à la capacité de réidentification d'une image offusquée par un classifieur.

2 Le réseau de neurones

Nous voulions effectuer nos essais sur ImageNet avec AlexNet, ce qui s'est avéré être pratiquement impossible à mettre en pratique du fait de limitations matérielles et de temps. Nous avons donc employé CIFAR-10 sur notre propre réseau de neurones implémenté à l'aide de Tensorflow et Keras et dont la structure est la suivante :

- Entrée $32 \times 32 \times 3$
- 32 convolutions 3×3 ($30 \times 30 \times 96$)
- Fonction d'activation ReLU
- Pooling max. 2×2 ($15 \times 15 \times 96$)
- 64 convolutions 3×3 ($14 \times 14 \times 288$)
- Fonction d'activation ReLU
- Pooling max. 2×2 ($7 \times 7 \times 288$)
- 64 convolutions 3×3 ($6 \times 6 \times 864$)
- Fonction d'activation ReLU
- Aplatissement en 64 caractéristiques
- Fonction d'activation ReLU
- n classes en fully-connected

Ce réseau fonctionne sans problème sous CIFAR-10. Sur ce dataset, la précision obtenue est souvent comprise entre 70% et 80%, avec un léger overfitting qui n'est pas très problématique dans notre cas (voir figure 1), car le plus important restera en priorité la capacité du modèle a évaluer des données de test obscurcies. Nous avons choisi un maximum de 30 époques, même si nous constatons que 10 suffisent. La librairie utilisée permet d'employer une condition d'arrêt en fonction de la perte au fil des époques.



(a) Précision/Époque (b) Matrice de confusion

FIGURE 1 – Analytiques de CIFAR-10 sur notre réseau de neurones

La raison pour laquelle nous spécifions ici n classes est que nous allons procéder à trois grands types de classification :

1. **Classification des données obscurcies sur un réseau entraîné par des données non-obscurcies**, ce qui permet de tester la capacité d'un réseau simple de classification à réidentifier l'image obscurcie. Cela nécessite donc $n = 10$ classes, ce qui est le nombre de classes présent dans CIFAR-10.
2. **Classification des données obscurcies sur un réseau entraîné par des données obscurcies et non-obscurcies**, permettant d'évaluer la force d'un tel réseau dans l'identification d'un contenu obscurci. On a également $n = 10$.
3. **Classification binaire des données obscurcies ou non-obscurcies**, nous souhaitons que notre classifieur reconnaisse au mieux si les données en entrée sont obscurcies ou non. Pour une telle classification binaire, on aura alors $n = 2$ classes.

Nous notons que pour les deux premiers types de classification, nous employons l'entropie croisée catégorique comme fonction de perte, ainsi qu'un softmax. La troisième classification utilise l'entropie croisée binaire comme fonction de perte, puis finalement la sigmoïde. La taille de batch reste fixe à 64 et l'optimiseur utilisé est ADAM.

La taille du dataset correspond à celle de CIFAR-10, notamment pour la première classification. C'est à dire 50 000 images d'entraînement et 10 000 images de validation. Dû à des limitations matérielles (car il faut obfusquer la moitié du dataset d'entraînement), nous sommes contraint d'utiliser seulement la moitié des données d'entraînement pour les deux dernières classifications, donc 25 000 images.

Dans ce compte-rendu, nous proposons deux approches supplémentaires à l'évaluation de l'obscuration :

— **Évaluation non-contextuelle** : on obscurcit et on évalue cette obscuration sur toute l'image, comme si nous étions uniquement intéréssé par la région obscurcie sans se soucier du reste. Il ne reste donc pas de zone non-obscurcie permettant d'en déduire plus aisément le contenu.

— **Évaluation contextuelle** : on obscurcit une région de l'image aléatoire, ce qui permet au modèle de s'adapter à tout type d'obscurcissement et potentiellement en déduire de l'information via le contenu non-obscurci. En l'occurrence, chaque région obscurcie possède une taille minimale de 16×16 et une position aléatoire dans l'image.

Ces évaluations sont alors menées sur nos quatres méthodes classiques d'obscuration de l'image, soit le **floutage**, le **masquage**, la **pixélisation** ainsi que le **chiffrement AES**. Pour chaque méthode, nous faisons varier les paramètres de la sorte, afin d'apprendre au maximum de notre modèle :

1. **Floutage** : La taille du noyau $k \in \{3, 5, 7, 9\}$.
2. **Masquage** : La couleur du masque $(r, g, b) \in [0, 255]^3$.
3. **Pixélisation** : La taille des pixels $T \in [4, 16]$.
4. **Chiffrement** : Une clé alphanumérique de 256 bits générée automatiquement.

Nous pouvons également faire varier les paramètres (x_1, y_1) et (x_2, y_2) de nos méthodes d'obscuration afin de sélectionner aléatoirement la taille de la zone que l'on souhaite obscurcir.

3 Classification de données obscurcies

3.1 Sans entraînement sur des donneés obscurcies

Dans un premier temps, nous entraînons notre modèle sur les 50 000 images d'entraînement de CIFAR-10. Nous obtenons dans le tableau 1 la précision obtenue par les différentes méthodes utilisées et dans des configurations différentes.

Ce tableau met en évidence les résultats obtenus et le potentiel de réidentification de notre modèle d'abord sur les images totalement obscurcies, avec lesquelles il est clair - et évident - que le modèle est incapable de classifier correctement les données de test lors d'un masquage ou d'un chiffrement car il n'existe pas de dépendance entre les données dont nous avons extrait les caractéristiques et des données bruitées ou uniformes. Nous constatons en revanche que le modèle est capable de reconnaître un quart du contenu à travers le flou ou la pixélisation.

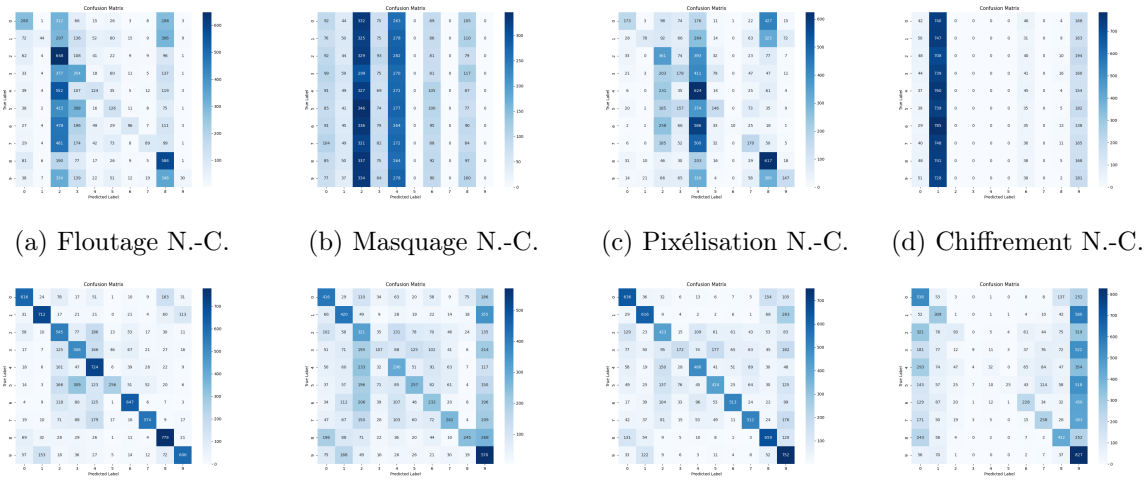
Nous voyons ensuite qu'introduire plus de contexte dans la classification permet à ce classifieur de reconnaître plus aisément le contenu obscurci. Nous déduisons premièrement pour le masquage et le chiffrement, que l'augmentation de la précision n'est pas un résultat d'une dépendance directe entre le contenu et le contenu obscurci, à l'inverse du floutage ou de la pixélisation qui contribuent à reconnaître plus de la moitié des images obscurcies.

Nous notons par ailleurs la présence d'un sur-apprentissage qui s'est accentué à cause des images ou régions obscurcies n'étant pas conformes à ce que le classifieur pouvait espérer à l'origine. Le nombre d'époques est relativement faible, et se situe aux alentours de 5 à 10 époques avant arrêt par Keras.

	Floutage	**Masquage**	**Pixélisation**	**Chiffrement**
Entraînement non-contextuel	24,26%	67,33%	64,09%	76,11%
Validation non-contextuelle	23,79%	10,10%	25,04%	10,10%
Entraînement contextuel	74,95%	68,50%	72,34%	75,86%
Validation contextuelle	59,67%	31,26%	51,93%	31,75%

TABLE 1 – Précision de la classification obtenue par chaque méthode

Dans la figure 2, nous pouvons voir que l'obscuration en elle-même a un faible potentiel de classification. Or, l'introduction de contexte dans la classification corrige cette erreur-ci, même si elle a tendance à introduire un certain biais vers une classe.

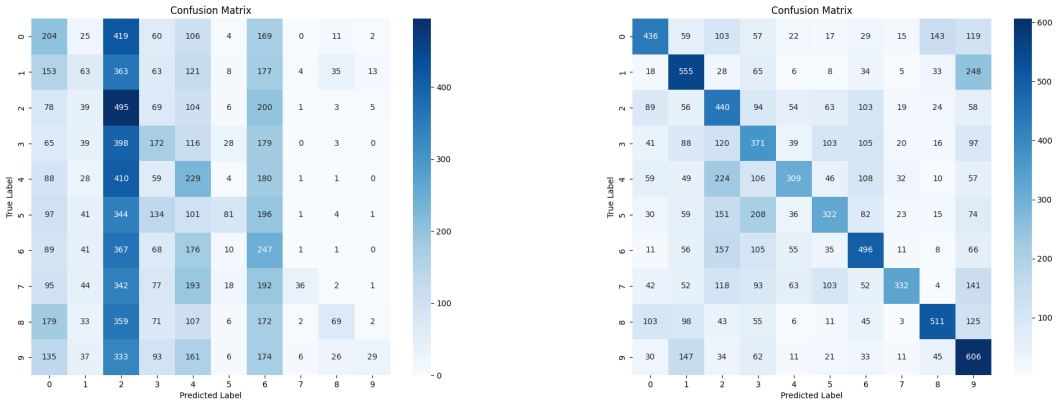


(a) Floutage N.-C. (b) Masquage N.-C. (c) Pixélisation N.-C. (d) Chiffrement N.-C.

(e) Floutage C. (f) Masquage C. (g) Pixélisation C. (h) Chiffrement C.

FIGURE 2 – Matrices de confusion obtenues lors de la classification

On veut donc voir si notre modèle est capable de classifier l'image lorsque toutes les méthodes d'obscuration sont appliquées alternativement sur nos images de test, ce qui divise les images obscurcies en 4 sous-groupes avec une méthode chacune. Nous obtenons alors une précision de 65,64% en entraînement et 16,25% ce qui effectue à peu près une moyenne des précisions précédentes lors d'un entrainement et d'une validation non-contextuels. Nous pouvons faire le même constat pour une classification contextuelle, avec 75,67% de précision d'entraînement et 43,78% ce qui est plutôt fort en sachant que le modèle sans obscuration possède une précision de test d'environ 71% sur CIFAR-10. La figure 3 met en évidence la capacité du modèle à classifier ces diverses formes d'obscuration suite à son entraînement et à l'ajout d'un contexte dans l'image.



(a) Obscuration non contextuelle (b) Obscuration contextuelle

FIGURE 3 – Matrices de confusion obtenues lors de la classification avec toutes les méthodes

3.2 Avec entraînement sur des donneés obscurcies

Nous souhaitons rendre notre classifieur plus robuste en l'entraînant sur 12 500 images inchangées et 12 500 images obscurcies. Les résultats sont présentés dans le tableau 2.

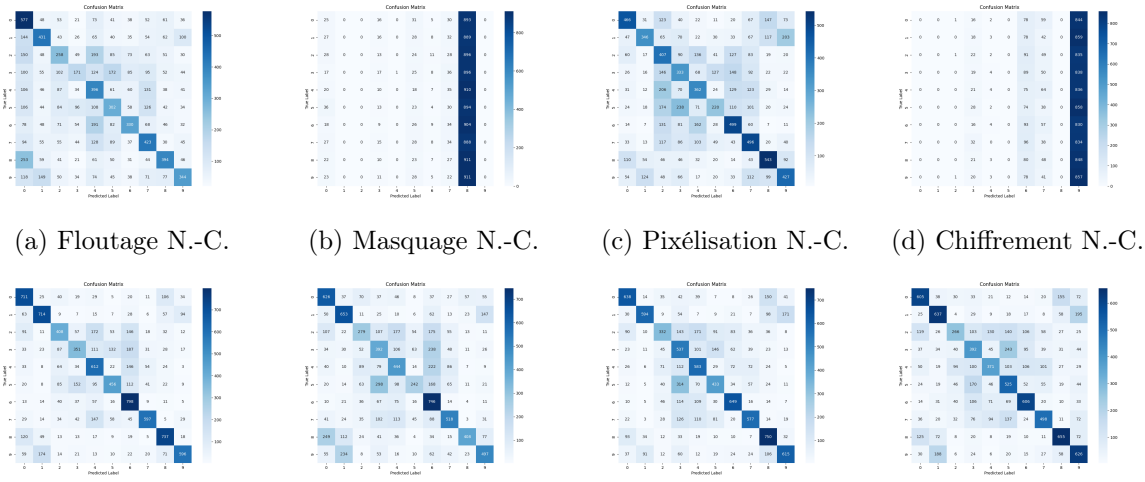
Nous obtenons de meilleurs résultats après 8-12 époques sur toutes les métriques de précision par rapport à notre précédente analyse, ce qui signifie que nous pouvons effectivement rendre plus robuste notre modèle de cette façon. Il existe certaines limites, comme par exemple pour le masquage ou le chiffrement qui ne varient pas fortement puisqu'il n'y a aucune similarité entre les classifications effectuées et ces données, mais également une limite que l'on peut constater avec le floutage, puisqu'à l'inverse de la pixélisation, l'introduction de contexte dans l'image n'a pas amélioré l'identification du contenu.

	Floutage	**Masquage**	**Pixélisation**	**Chiffrement**
Entraînement non-contextuel	57,37%	38,24%	60,24%	37,36%
Validation non-contextuelle	36,26%	10,19%	40,99%	10,12%
Entraînement contextuel	72,96%	67,54%	69,25%	70,16%
Validation contextuelle	59,80%	48,05%	57,08%	51,81%

TABLE 2 – Précision de la classification obtenue par chaque méthode

Dans une scénario non-contextuel, il est clair que nous obtenons à présent des résultats de classification plus cohérents en ce qui concerne la classification de nos données obscurcies pour un floutage ou une pixélisation, ce que nous observons dans la figure 4. En revanche, pour un masquage ou un chiffrement, on obtient un biais extrême vers une classe.

Ce biais est alors corrigé par l'introduction de contexte dans la classification, ce qui corrige légèrement aussi le problème que nous avions dans la précédente section, où une classe avait un plus fort biais que nous avons actuellement.

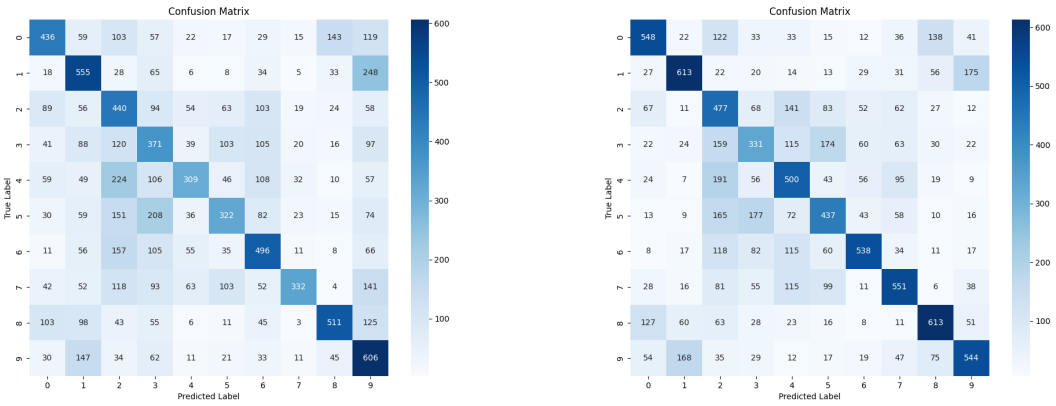


(a) Floutage N.-C. (b) Masquage N.-C. (c) Pixélisation N.-C. (d) Chiffrement N.-C.

(e) Floutage C. (f) Masquage C. (g) Pixélisation C. (h) Chiffrement C.

FIGURE 4 – Matrices de confusion obtenues lors de la classification

Comme nous avons fait dans notre première analyse, nous alternons les différentes méthodes d'obscuration dans le modèle. Nous obtenons une précision de 53,23% en entraînement et 23,15% en validation, ce qui est largement supérieur que nos précédents résultats sans introduction de contexte. Tout de même, on améliore cela par l'ajout d'un contexte, et on obtient 67,24% de précision en entraînement et 51,52% en validation. Ce qui est d'ailleurs plutôt intéressant car non seulement on est à présent capable de classifier correctement plus de la moitié des images même après obscuration, on a également réduit l'overfitting ! La figure 5 montre cette amélioration, qui reste légèrement faible, mais corrige un certain biais.



(a) Obscuration non contextuelle

(b) Obscuration contextuelle

FIGURE 5 – Matrices de confusion obtenues lors de la classification avec toutes les méthodes

4 Détection de l'offuscation

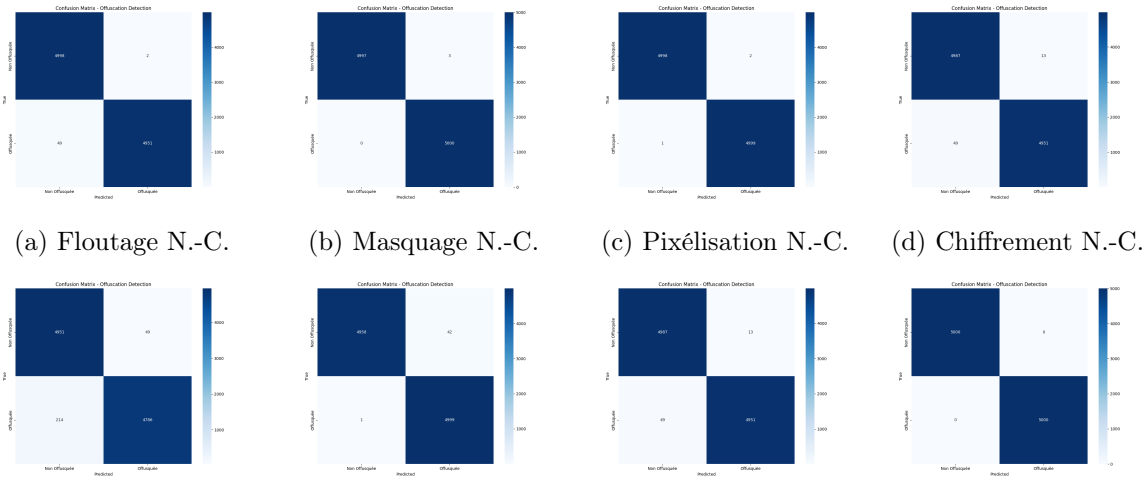
À présent, nous mettons en place le même réseau de neurones avec en sortie deux classes : offusquées ou non offusquée. De la même façon que précédemment, nous obscurcissons la moitié du dataset soit 12 500 images et gardons l'autre intacte. Nous faisons de même avec les données de test, soit 5 000 images obscurcies et 5 000 inchangées.

Le potentiel de reconnaissance mis en évidence dans la figure 3 est impressionnant, voire peut être suspicieux dans une certaine mesure, mais il est important de se souvenir du contexte dans lequel nous effectuons cette analyse. En effet, les images naturelles ne possèdent pas toujours les mêmes qualités qu'une image offusquée par nos diverses méthodes, et le floutage - qui possède la précision la plus basse - est parmi celles-ci la moins dégradante ; et pourtant la précision ne descend pas en dessous de 97%. Nous voyons que notre modèle est capable d'assimiler à une image en entrée si la méthode utilisée lors de l'entraînement a été appliquée ou non sur l'image de test avec certitude. Le nombre d'époques s'arrêtait à 30, ce qui signifie qu'il peut être possible d'avoir de résultats encore meilleurs.

	Floutage	**Masquage**	**Pixélisation**	**Chiffrement**
Entraînement non-contextuel	99,77%	99,57%	99,97%	100,00%
Validation non-contextuelle	99,49%	99,97%	99,97%	100,00%
Entraînement contextuel	98,62%	99,70%	99,65%	100,00%
Validation contextuelle	97,37%	99,57%	99,38%	100,00%

TABLE 3 – Précision de la reconnaissance obtenue par chaque méthode

Les matrices de confusion dans la figure 6 possèdent toutes le même motif, avec une très légère erreur sur nos méthodes sauf sur le chiffrement, qui permet une reconnaissance parfaite de l'image offusquée.



(a) Floutage N.-C. (b) Masquage N.-C. (c) Pixélisation N.-C. (d) Chiffrement N.-C.

(e) Floutage C. (f) Masquage C. (g) Pixélisation C. (h) Chiffrement C.

FIGURE 6 – Matrices de confusion obtenues lors de la reconnaissance

5 Discussion sur nos résultats

Nous avons eu des résultats très satisfaisants lors de notre étude, et intéressants puisque nous pouvons faire directement le lien entre la détection de l'offuscation - qui est très bonne - et sa réidentification, qui possède ses propres atouts. Cela nous permettrait éventuellement de considérer un autre réseau de neurones pouvant détecter et classifier le type d'obscuration parmi les méthodes utilisées. Par ailleurs, cela pourrait également nous pousser davantage à explorer d'autres méthodes d'obscuration et les employer comme nous l'avons fait avec celles présentées dans ce compte-rendu.