Note méthodologique : preuve de concept

Dataset retenu

Le **Stanford Dogs Dataset** est un ensemble de données utilisé pour la classification d'images, spécifiquement conçu pour la reconnaissance de races de chiens. Ce dataset a été créé par l'Université de Stanford et contient environ **20 580 images** réparties sur **120 races de chiens différentes**. Ces images ont été collectées à partir de sources variées, telles que le web, et sont de résolutions et formats différents, ce qui en fait un excellent jeu de données pour les applications d'apprentissage en profondeur, en particulier pour l'entraînement de modèles de classification d'images.

Les images sont classées par dossier, chaque dossier correspondant à une race spécifique de chien. Chaque image est étiquetée avec la race à laquelle elle appartient, et les classes sont organisées selon la taxonomie de la race de chien.

Détails techniques :

- **Nombre de classes** : 120 races de chiens, couvrant une large diversité de chiens de toutes tailles et formes.
- **Type de données** : Images (JPEG).
- **Taille des images** : Les images ont des tailles variables, mais elles sont généralement redimensionnées pour les besoins de l'apprentissage en profondeur.
- **Format d'étiquette** : Chaque image est étiquetée avec une seule classe correspondant à une race spécifique.

Le Stanford Dogs Dataset est utilisé pour la reconnaissance d'objets et pour les tâches de classification d'images dans des modèles d'apprentissage automatique et d'apprentissage profond, notamment les réseaux de neurones convolutifs (CNN). Il sert aussi de base pour évaluer les performances des modèles de vision par ordinateur dans des conditions réelles.

Applications :

- **Classification d'images** : Utilisé pour entraîner des modèles de vision par ordinateur pour reconnaître et classer les races de chiens.
- **Recherche en vision par ordinateur** : L'ensemble de données est couramment utilisé pour tester de nouvelles architectures de modèles et des techniques d'optimisation dans le domaine de la vision par ordinateur.

Ce dataset est un choix populaire pour les chercheurs et les développeurs qui souhaitent entraîner et tester des modèles de classification d'images de manière pratique et sur un jeu de données riche et varié.

Les concepts de l'algorithme récent

RandAugment est une méthode d'augmentation de données utilisée principalement dans le domaine de la vision par ordinateur pour améliorer la performance des modèles d'apprentissage automatique. Elle s'inscrit dans une lignée d'approches visant à enrichir les ensembles de données d'entraînement en appliquant des transformations aléatoires aux images. Le but est de rendre les modèles plus robustes en leur fournissant une plus grande variété d'exemples sans nécessiter une augmentation massive du volume des données. RandAugment se distingue par sa simplicité, sa flexibilité et son efficacité, tout en ne nécessitant qu'un nombre réduit de paramètres pour ajuster les transformations appliquées.

1. Contexte et Motivation

Dans le cadre des tâches de classification d'images, l'augmentation de données est une stratégie couramment utilisée pour éviter le surapprentissage, ou *overfitting*, en introduisant de la variabilité dans les données d'entraînement. Traditionnellement, des transformations comme la rotation, le retournement horizontal ou l'ajustement de la luminosité sont appliquées pour diversifier les images d'entraînement. Cependant, ces méthodes peuvent nécessiter des ajustements fins et des choix de transformations manuels.

RandAugment propose une solution plus flexible et automatisée en choisissant aléatoirement parmi un ensemble de transformations de manière contrôlée, ce qui permet de maximiser la diversité des données sans nécessité de définir de manière explicite les types de transformations à appliquer. Cela simplifie le processus et rend l'augmentation des données plus accessible et plus rapide.

2. Les Principes de RandAugment

Le principe fondamental de RandAugment repose sur l'application aléatoire de transformations géométriques et colorimétriques aux images. Contrairement à des méthodes plus complexes comme *AutoAugment* (qui cherche à apprendre un ensemble optimal de transformations), RandAugment se distingue par sa simplicité. Son approche repose sur deux hyperparamètres principaux : le nombre de transformations à appliquer et l'intensité de chaque transformation.

a) Choix aléatoire des transformations :
RandAugment choisit de manière aléatoire parmi une liste de transformations de base qui incluent des opérations telles que la rotation, la coupure, le changement de contraste, la saturation, ou encore la translation. Ces transformations sont choisies selon une probabilité définie à l'avance, ce qui permet une grande diversité dans les images générées.

b) Contrôle de l'intensité des transformations :
Chaque transformation appliquée à une image possède un paramètre d'intensité. Ce paramètre détermine l'ampleur de la transformation : par exemple, la rotation peut varier de 0 à 30 degrés, ou le changement de contraste peut être modéré ou plus intense. L'intensité est également choisie aléatoirement pour chaque transformation, permettant ainsi une grande variété dans les exemples générés.

c) Augmentation à un seul niveau :
Contrairement à des méthodes plus complexes comme *AutoAugment*, RandAugment ne recherche pas un ensemble optimal de transformations à appliquer, ce qui réduit la charge computationnelle. Au lieu de cela, il se contente d'une simple combinaison de

transformations appliquées de manière aléatoire, ce qui le rend beaucoup plus rapide et plus simple à mettre en œuvre.

3. Applications et Avantages

RandAugment est principalement utilisé dans les tâches de classification d'images, notamment pour améliorer la performance des réseaux neuronaux sur des ensembles de données limités. En générant une plus grande variété d'exemples à partir des données initiales, cette méthode aide les modèles à mieux généraliser et à réduire le risque de surapprentissage.

Les avantages de RandAugment sont nombreux :

- **Simplicité d'utilisation :** L'algorithme ne nécessite que peu de réglages et d'hyperparamètres, ce qui le rend facile à utiliser et à implémenter.
- **Efficacité :** Par rapport à d'autres méthodes comme *AutoAugment*, RandAugment est beaucoup moins coûteux en termes de calcul, car il ne nécessite pas de recherche complexe pour trouver l'ensemble optimal de transformations.
- **Amélioration de la généralisation :** En augmentant la diversité des données d'entraînement, RandAugment permet au modèle de mieux se généraliser à des données qu'il n'a pas vues pendant l'entraînement, ce qui est crucial dans des scénarios de production où de nouvelles données sont constamment introduites.
- **Robustesse :** Les transformations aléatoires aident le modèle à devenir plus robuste aux variations naturelles qui peuvent survenir dans les données réelles, comme les changements d'éclairage, d'angle, ou d'orientation.

4. Conclusion

RandAugment est une approche d'augmentation de données simple et efficace qui permet de renforcer les modèles d'apprentissage automatique, en particulier dans le cadre de la classification d'images. En appliquant des transformations aléatoires avec des niveaux d'intensité variés, cette méthode enrichit les ensembles de données sans nécessiter de calculs complexes ou de réglages fins des hyperparamètres. Grâce à sa simplicité et à son efficacité, RandAugment est une solution très populaire dans les projets de vision par ordinateur, où la diversité des données est cruciale pour la performance du modèle.

La modélisation

Avant d'entamer la description de la méthodologie et la phase de modélisation, il convient de détailler au préalable la manière dont les données en entrée ont été préparées et traitées. Le processus s'appuie sur un dataset d'origine contenant 3 races de chiens (chihuahua, malinois, malamute), soit un total de 480 images. Ce dataset a été enrichi grâce à deux méthodes distinctes de data augmentation, générant ainsi un total de trois datasets. Ces derniers, contenant chacun 3 races de chiens pour un total de 9600 images, ont servi à entraîner respectivement trois modèles CNN construits from scratch avec une structure identique et les mêmes paramètres. Par ailleurs, les deux datasets augmentés contiennent un nombre équivalent d'images (9600 images), garantissant une modélisation homogène et des résultats plus cohérents.

Informations sur les jeux de données :

- Dataset d'origine : 3 races de chiens, 480 images au total.
- Dataset manuellement data augmenter : 3 races de chiens, 9600 images au total.
- Dataset data augmenter avec RandAugment : 3 races de chiens, 9600 images total.

Concernant le dataset d'origine, ce dernier a subi un pré-traitement incluant plusieurs transformations essentielles comme notamment :

- Redimensionnement des images en 224x224
- Application d'une Bounding Box permettant de capturer et de centrer le sujet de l'image
- Transformation de la colorimétrie, passant de 3 canaux RGB a 1 canal d'échelle de gris, allégeant ainsi les modélisations futures.

Concernant le dataset manuellement data augmenter, ce dernier étant une variante du dataset d'origine. Il bénéficie du pré-traitement effectuer et a subi plusieurs autres transformations avec l'utilisation de la méthode « ImageDataGenerator » de la librairies Keras :

- Rotation aléatoire jusqu'à 40 degrés.
- Déplacement horizontale et vertical jusqu'à 20% de la largeur et hauteur de l'image.
- Distorsion géométrique et transformation de cisaillement jusqu'à 20%.
- Zoom aléatoire jusqu'à 20%.
- Inversement horizontale et vertical aléatoire de l'image.
- Ajustement de la luminosité allant de 50% à 150%.
- Remplissage des pixels vides génères par des transformations avec les pixels les plus proches.
- Décalage des canaux de couleurs jusqu'à 150%, modifiant les teintes.

Concernant le dataset augmenter avec « RandAugment », ce dernier étant une variante du dataset d'origine. Il bénéficie du pré-traitement effectuer et a subi plusieurs autres transformations avec l'utilisation de la méthode « RandAugment » de la librairie Torchvision, appliquant ainsi aléatoires de la data augmentation en respectant les 2 paramètres suivant :

- N : Nombre d'opérations de data augmentation, ici égal à 5.
- M : Magnitude, c'est-à-dire la densité des transformations, ici égal à 7 pour générer des images nouvelles tout en gardant un aspect réaliste.

Trois modélisations ont ainsi été effectuer en respectant le même nombre d'hyperparamètres, de couches, les mêmes paramètres de compilation. Voici en détails les informations du modèle utiliser :

Réseaux de neurones convolutionels :

- 1 couche de normalisation des valeurs des pixels des images entre 0 et 1 en divisant par 255, facilitant l'apprentissage.
- 4 couches de convolution appliquant 128, 64 ,32 ,16 filtres sur des zones de 3x3 pixels. La fonction ReLu introduit de la non-linéarité.
- 4 couches de réduction de la taille des caractéristiques. Diminuant la complexité et le risque de surapprentissage.
- 4 couches de désactivation aléatoire de 30% des neurones à chaque itération pour éviter le surapprentissage.
- 1 couche de transformation d'aplatissement passant de vecteur 2D à 1D.
- 1 couche entièrement connectée avec 64 neurones pour l'apprentissage des combinaisons complexes des caractéristiques.
- 1 couche de sortie Softmax produisant des probabilités pour chaque classe.
- La compilation du modèle avec un optimisateur Adaptive Moment Estimation (ADAM) combinant les avantages des descentes de gradient classiques et adaptatives. Une fonction de perte « SparceCategoricalCrossentropy » utiliser pour les problèmes de classification multi-classes. La métrique « Accuracy » qui surveille la précision du modèle pendant l'entrainement permettant de mesurer la performance des prédictions correctes.

Les métriques retenues pour l'évaluation de ce modèle, comme indiqué ci-dessus sont donc :

- Accuracy : Mesure la performance des prédictions correctes.
- La fonction de perte de type SparceCategoricalCrossentropy : Mesure l'écart entre les prédictions du modèle et les labels cibles lors d'une classification multi-classes. Elle guide le modèle en indiquant a quel points les prédictions actuelles sont incorrectes pour ajuster ses paramètres pour affiner les prédictions.

3 modélisations ont ainsi été effectuer avec 10 epochs et un batch size de 32, respectivement comme suit :

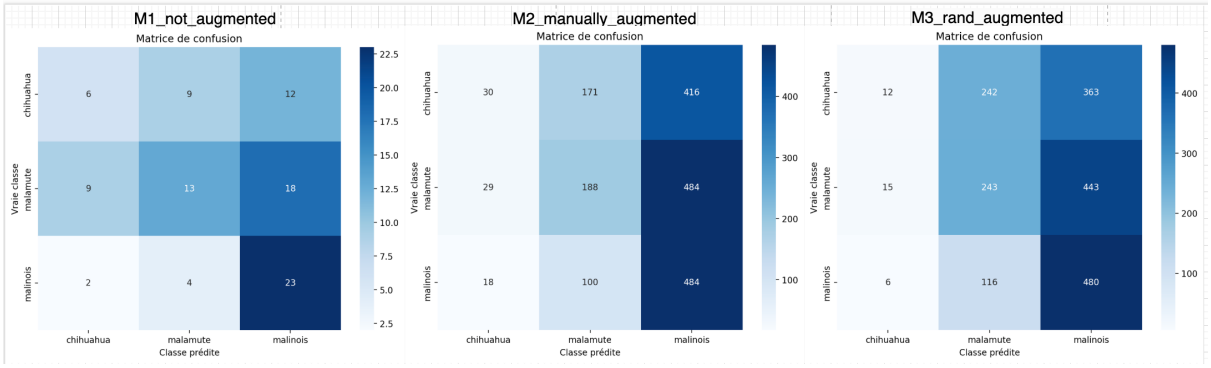
- M1_not_augmented : Entrainer avec le dataset d'origine,
- M2_manually_augmented : Entrainer avec le dataset manuellement data augmenter.
- M3_rand_augmented : Entrainer avec le dataset data augmenter avec RandAugment.

Une synthèse des résultats

La modélisation de ces 3 modèles à permit d'obtenir des résultats cohérents avec la faible diversité et densité de données.

Voici une synthèse des résultats obtenues :

M1_not_augmented			
Accuracy	Loss	Validation Accuracy	Validation Loss
0.4948	1.0032	0.4375	1.0569
M2_manually_augmented			
Accuracy	Loss	Validation Accuracy	Validation Loss
0.4345	1.0353	0.3656	1.0821
M3_rand_augmented			
Accuracy	Loss	Validation Accuracy	Validation Loss
0.4177	1.0465	0.3771	1.0797



Observations :

- **M1_not_augmented** reste le meilleur modèle en termes de précision brute, mais son risque de sur-apprentissage limite sa généralisation.
- L'augmentation des données (M2 et M3) n'a pas encore permis d'améliorer significativement les performances. Cependant, elle a légèrement réduit le sur-apprentissage observé avec M1.
- Les matrices de confusion montrent que les trois modèles ont des difficultés à distinguer correctement la classe "Chihuahua".
- **M3_rand_augmented** est légèrement meilleur que **M2_manually_augmented** en termes de capacité à généraliser, grâce à la diversité accrue apportée par les augmentations aléatoires.
- **M2_manually_augmented** souffre d'un biais potentiel introduit par des augmentations moins variées et dépendantes des règles prédéfinies.

L’analyse de la feature importance globale et locale du nouveau modèle

Un test a été réalisé afin de vérifier la pertinence des deux hyperparamètres clés de la méthode RandAugment : N, qui représente le nombre d’opérations de transformation appliquées, et M, qui correspond à l’intensité de ces transformations. L’objectif de ce test est de comprendre l’impact de ces paramètres sur les performances du modèle en termes de précision, de perte, et de validation. Pour cela, trois configurations distinctes ont été mises en place, chacune utilisant un ensemble de données généré avec des valeurs spécifiques de N et M. Ces expérimentations permettent d’évaluer dans quelle mesure ces paramètres influencent la qualité des données augmentées et les métriques du modèle, afin de déterminer les réglages les plus adaptés pour une amélioration optimale.

RandAugment, une méthode de data augmentation, repose sur deux hyperparamètres clés :

- **N** : le nombre d’opérations de transformation appliquées.
- **M** : la magnitude (intensité) des transformations effectuées.

Ces hyperparamètres influencent directement les caractéristiques des données augmentées et, par conséquent, les performances des modèles entraînés avec ces données. Afin d’évaluer leur impact, trois configurations de valeurs pour N et M ont été testées, et leurs résultats comparés en termes de précision (accuracy), perte (loss), précision de validation (validation accuracy) et perte de validation (validation loss).

Test_1 : N = 1, M = 3			
Accuracy	Loss	Validation Accuracy	Validation Loss
0.9931	0.0222	0.9958	0.0203
Test_2 : N = 5, M = 7			
Accuracy	Loss	Validation Accuracy	Validation Loss
Test_3 : N = 10, M = 20			
Accuracy	Loss	Validation Accuracy	Validation Loss
0.3812	1.0897	0.3510	1.1013

Les tests démontrent que les hyperparamètres N et M ont un impact significatif sur les performances des modèles entraînés avec RandAugment. Des valeurs excessives peuvent engendrer des données non représentatives ou bruitées, réduisant ainsi les métriques de performance. Une calibration minutieuse de ces hyperparamètres est essentielle pour maximiser les bénéfices de la data augmentation.

Les résultats montrent que le **Test_1** (N = 1, M = 3) a produit d’excellentes métriques, avec une précision et une précision de validation proches de 1, ainsi que des pertes très faibles. Cependant, les prédictions du modèle se révèlent très médiocres, car celui-ci prédit systématiquement la même classe, indiquant un surajustement potentiel ou une incapacité à généraliser correctement. Ces observations soulignent la nécessité d’un équilibre dans les choix des hyperparamètres pour garantir des données augmentées de qualité et des modèles performants.

Les limites et les améliorations possibles

L'approche RandAugment présente plusieurs avantages pour la data augmentation, mais également certaines limites qui impactent les performances et l'interprétabilité des modèles basés sur des réseaux neuronaux convolutionnels (CNN). Ces limites, ainsi que des axes d'amélioration, sont présentés ci-dessous.

Limites :

1. **Manque de contrôle sur les transformations appliquées :**

RandAugment applique des transformations de manière aléatoire, sans tenir compte de la nature spécifique des données ou des objectifs du modèle. Cela peut générer des augmentations non pertinentes, introduisant du bruit et affectant la qualité de l'apprentissage.

2. **Sur- ou sous-paramétrage des hyperparamètres** N **et** M **:**

Comme le montrent les tests réalisés, des valeurs trop faibles (Test_1) peuvent entraîner une homogénéité excessive dans les données augmentées, menant à des modèles biaisés (prédiction constante d'une seule classe). À l'opposé, des valeurs trop élevées (Test_3) introduisent une trop grande variabilité, réduisant les performances globales du modèle.

3. **Coût computationnel :**

L'augmentation des données, en particulier avec des paramètres N et M élevés, engendre un coût computationnel significatif en termes de temps et de ressources pour générer les données augmentées et entraîner le modèle.

4. **Manque d'interprétabilité des résultats :**

La nature aléatoire de RandAugment complique l'analyse des contributions spécifiques de chaque transformation aux performances du modèle, limitant l'interprétabilité.

Améliorations envisageables pour gagner en performance et interprétabilité :

1. **Augmenter la variabilité des données générées :**

   • Générer un plus grand nombre d'images augmentées avec des paramètres N et M légèrement plus élevés, tout en veillant à ne pas introduire un excès de bruit. Cela permettrait d'enrichir les données d'apprentissage et de mieux généraliser le modèle.

2. **Optimisation des hyperparamètres :**

   • Mettre en place une recherche systématique des hyperparamètres N et M via des techniques comme la recherche aléatoire ou Bayesienne pour trouver la meilleure combinaison adaptée aux données spécifiques.

3. **Optimisation des hyperparamètres du modèle CNN :**

   • Ajuster les couches convolutionnelles, le taux d'apprentissage, les fonctions d'activation, et les méthodes de régularisation pour améliorer les performances globales. La compilation du modèle pourrait également être optimisée en ajustant les fonctions de perte et les métriques utilisées.

4. **Stratégie de transformation conditionnelle :**

   • Introduire une version améliorée de RandAugment où les transformations sont conditionnées sur les caractéristiques de l'image ou des classes cibles, permettant ainsi de générer des augmentations plus pertinentes.

5. **Evaluation de la qualité des images générées :**

   • Intégrer une étape de validation des données augmentées en utilisant un score de similarité ou des métriques perceptuelles pour garantir la pertinence des transformations appliquées.

6. **Réduction du coût computationnel :**

   • Mettre en place des pipelines d'augmentation par lot, ou utiliser des frameworks optimisés (comme TensorFlow Data) pour accélérer le processus d'augmentation des données.