INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

Projet d'apprentissage faiblement supervisé

FixMatch

Yousra Bouchikhi, Hamza Belkarkor

2^{me} année en Modélisation et Intelligence Artificielle, double diplôme
à l'INSA Toulouse et l'ENSEEIHT - Toulouse, France

Juin 2023

ModIA
Promotion 2024

2022 - 2023

Sommaire

1 Introduction

FixMatch est un algorithme d'apprentissage semi-supervisé qui vise à améliorer les performances des modèles d'apprentissage automatique en tirant parti à la fois de données étiquetées et non étiquetées. Il a été proposé comme une méthode efficace pour entraîner des modèles avec des ressources de données limitées.

L'idée principale de FixMatch repose sur un processus en deux étapes. Tout d'abord, un petit sous-ensemble d'échantillons étiquetés est utilisé pour entraîner initialement le modèle. Ensuite, le modèle est utilisé pour prédire les étiquettes des échantillons non étiquetés, et seules les prédictions les plus fiables sont considérées comme étant correctes.
Pour renforcer la fiabilité des prédictions, FixMatch utilise une technique appelée "consistency regularization". Cela consiste à appliquer une augmentation de données aux échantillons non étiquetés, puis à comparer les prédictions du modèle sur les versions augmentées et non augmentées de ces échantillons. Les différences entre les prédictions sont utilisées pour calculer une perte supplémentaire, ce qui permet d'encourager la cohérence des prédictions.
FixMatch combine ensuite les échantillons étiquetés et non étiquetés dans un ensemble d'entraînement augmenté, et utilise une perte combinée pour entraîner le modèle. Ce processus est itérativement répété pour améliorer progressivement les performances du modèle.

Le projet a pour but d'observer des résultats sur l'algorithme FixMatch.
Nous commençons par entraîner en mode supervisé, un modèle de réseau de neuronnes convolutifs 2D fourni sur les données CIFAR10. Nous évaluons les résultats en prenant toutes les données CIFAR10 (50.000 images), puis sur 10% de ces données , ensuite 5% et 1% des données en utilisant une augmentation des données.
La deuxième partie de ce projet porte sur l'implémentation de l'algorithme FixMatch en mode semi-supervisé en utilisant l'article fourni, ensuite nous testons les performances de cet algorithme en utilisant seulement 10% des données labellisées, puis 5% et 1% avec une augmentation faible et forte des données.
Nous finissons par comparer les performances de l'algorithme FixMatch en mode semi-supervisé avec les résultats obtenus en mode supervisé dans le cas où on ne choisit que 10% des données puis 5% et 1%.

2 RandAugment

L'augmentation (forte) utilisée le long de cette étude est **RandAugment**, qui est une méthode qui applique une combinaison aléatoire de transformations à une image, contrôlée par les paramètres n et m. Pour mettre en œuvre RandAugment, nous utilisons la bibliothèque **imgaug.augmenters** et définissons une instance de la classe **RandAugment** avec des valeurs spécifiques de $n = 2$ et $m = 9$.

3 Apprentissage supervisé

3.1 Toutes les données

Durant toute la suite du projet nous utilisons le modèle de réseau de neuronnes convolutifs 2D d'architecture la suivante :

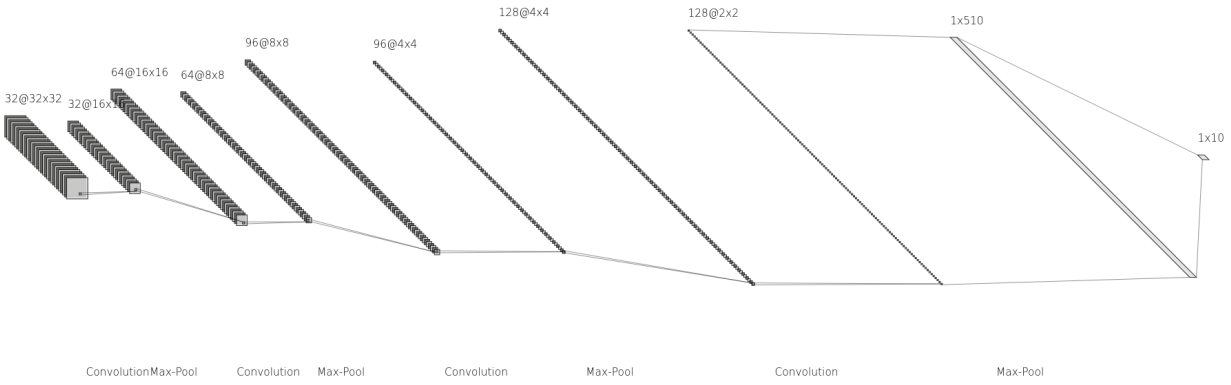


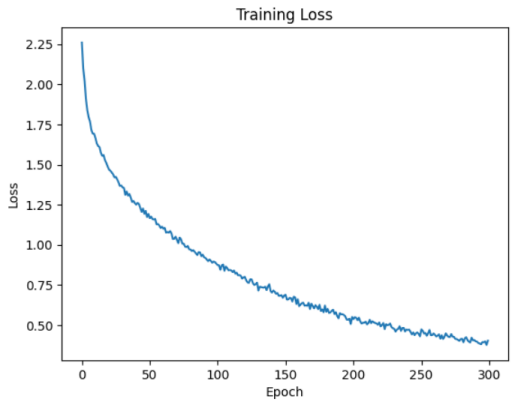
Figure 1: Architecture du réseau CNN

Nous commençons par entraîner le modèle avec toutes les données CIFAR10 (50000, 32, 32, 3). Les résultats obtenus sont satisfaisants.
Nous atteignons une perte de 0.15 et une précision de 0.96 sur les données d'entrainement, ainsi qu'une perte de 4.46 et une précision de 0.71 sur les données de test. On peut déjà remarquer qu'il y a un peu de surapprentissage vu l'écart de précision de 0.25 entre les données d'entraînement et de test.

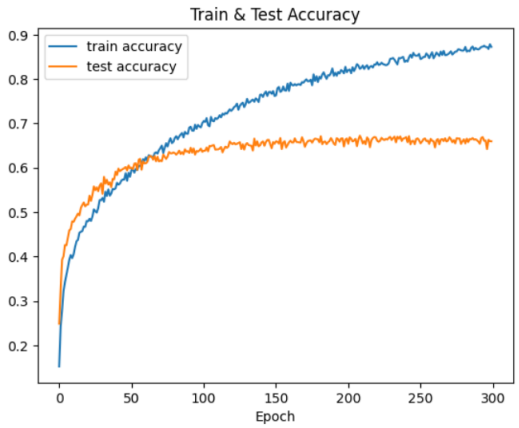
3.2 10% des données

Dans cette partie, nous choisissons 10% des données d'entrainement pour représenter notre nouvelle base de données d'entraînement du modèle. Les données sont choisies de façon à avoir une bonne répartition des 10 classes sur les données grâce au paramètre $stratify$ de la fonction $train\_test\_split$ de la librairie $sklearn$.

Au bout de 300 epochs, nous obtenons les résultats suivants:



(a) Fonction perte d'entraînement



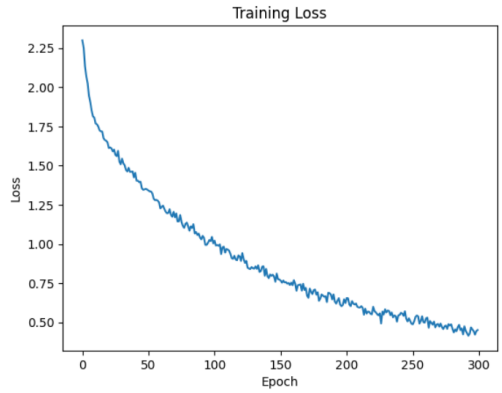
(b) Précision d'entraînement et de test

Figure 2: Métriques pour 10% des données

On remarque que l'on a pas beaucoup perdu en précision test par rapport à l'entraînement avec la totalité des données (0.65 contre 0.71)

3.3 5% des données

De la même manière, nous choisissons une nouvelle base de données construite ici de seulement 5% des données d'entrainement totales.
Au bout de 300 epochs nous obtenons les résultats suivants :



(a) Fonction perte d'entraînement

(b) Précision d'entraînement et de test

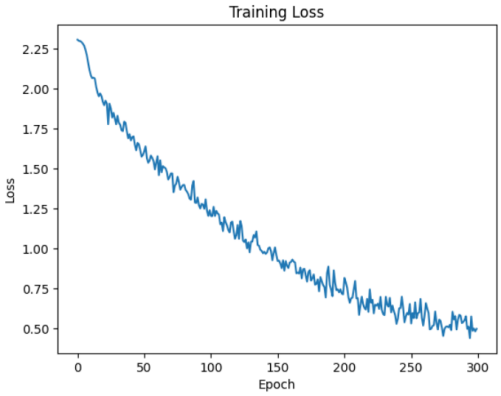
Figure 3: Métriques pour 5% des données

On remarque que l'on a perdu en précision de test (0.46) par rapport aux cas précédents, ce qui est normal puisque 5% des données ne sont pas suffisantes pour que le modèle assimile les différentes caratéristiques de nos données.

3.4 1% des données

Finalement, nous reconstruisons une nouvelle base de données d'entraînement avec seulement 1% des données, de la même manière.

Après 300 epochs d'entraînement nous obtenons les résultats suivants :



(a) Fonction perte d'entraînement

(b) Précision d'entraînement et de test

Figure 4: Métriques pour 1% des données

Nous remarquons une perte d'avantage de précision test (0.4) par rapport aux cas précédents. Il y a aussi du surapprentissage, puisque la perte d'entrainement ne cesse de diminuer mais les performances du modèle sur les données de test stagnent au bout de 120 epochs.

4 Apprentissage semi-supervisé

Dans la partie 3 de ce projet, nous abordons l'apprentissage semi-supervisé en implémentant l'algorithme FixMatch.

4.1 FixMatch: version simplifiée

Dans cette partie du projet, nous simplifions l'algorithme FixMatch en nous concentrant sur les principales étapes et en ajustant certains paramètres pour faciliter l'implémentation.

Voyons comment FixMatch est appliqué en pratique. Le pipeline global est résumé par la figure suivante:

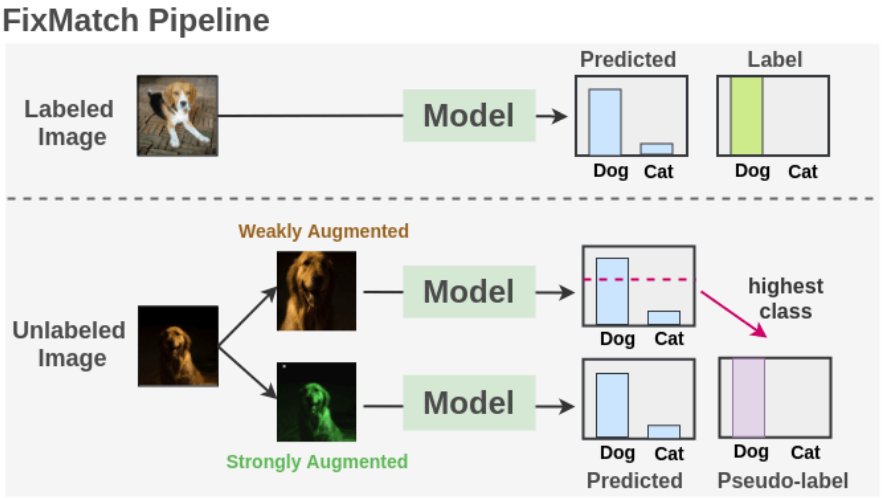


Figure 5: Pipeline FixMatch
Image source: https://amitness.com/2020/03/fixmatch-semi-supervised/

- L'augmentation forte utilisée est celle présentée dans la partie 2.
- L'augmentation faible utilisée est celle introduite par le papier: **Random Horizontal Flip**.
- Optimiseur: **Adam** ($learning\_rate = 10^{-4}$)
- Pour la partie supervisée de la perte: **Sparse Categorical Crossentropy**.
- Pour la partie non-supervisée: nous avions codé notre fonction de perte avec $\tau = 0.95$.
- Métriques d'entraînement et de test: **Sparse Categorical Accuracy**
- $\lambda_u = 1$
- $\mu = 2$: Cela signifie que nous utilisons deux fois plus d'images non étiquetées que d'images étiquetées.
- $B = 10$ pour 5 et 10% des données labellisées et $B = 64$ pour 1%.

4.2 Choix de données

Afin d'obtenir un ensemble de données labellisées et non labellisées uniforme pour pouvoir comparer nos résultats, nous avons implémenter une fonction $generate\_data\_cifar$ qui utilise la méthode $train\_test\_split$ de Sickit Learn avec $random\_state = 2$. On s'assure également que le nombre d'images par classe est le même en modifiant le paramètre $stratify$ de $train\_test\_split$.
Dans le cas où on ne garde que 10% de CIFAR-10 par exemple, on s'assure qu'on a bien 500 images par classe:

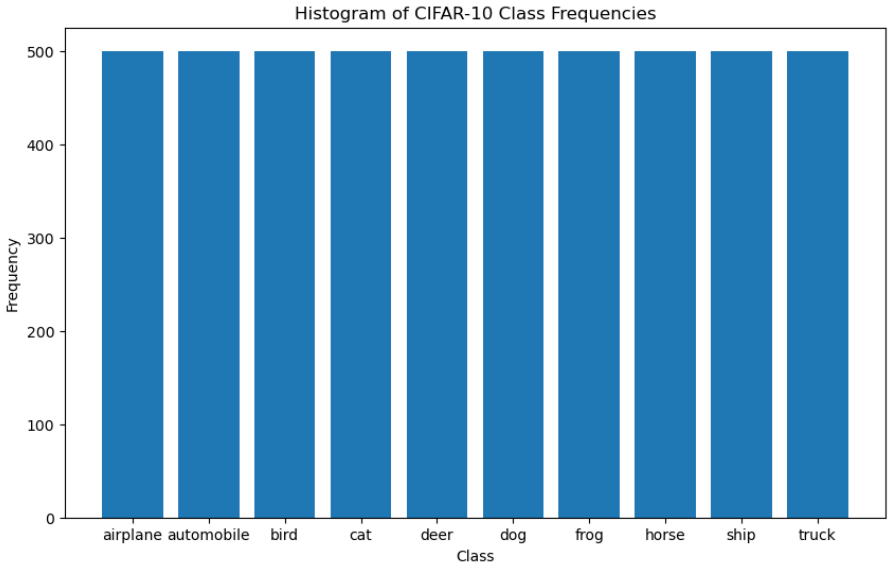


Figure 6: Histogramme de distribution dans le cas de 10% des données labellisées

4.3 Résultats

Ayant constaté que l'apprentissage semi-supervisé nécessite plus d'epochs pour apprendre, nous prenons un nombre d'epochs égal à 800 pour les cas de 10 et 5%, et égal à 2000 pour le cas 1% où l'apprentissage nécessite encore plus d'epochs afin de pouvoir comparer.

4.3.1 10% de données labellisées

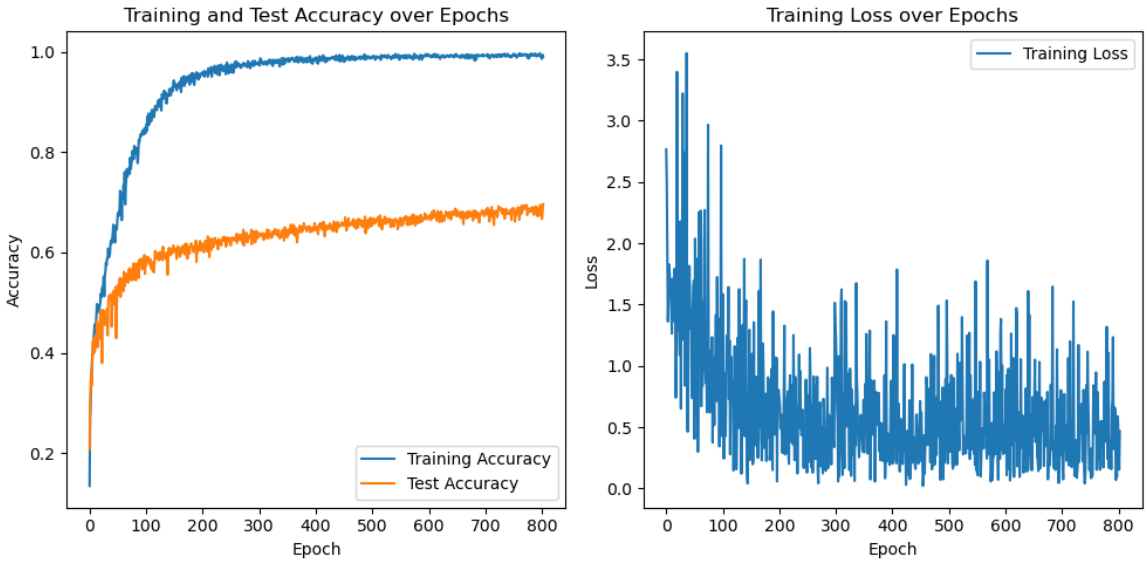


Figure 7: Précisions d'entrainement/de test et la loss au cours des epochs

Dans cette situation, il est observable que la précision de test atteint 69%, et elle est susceptible d'augmenter en augmentant le nombre d'epochs (on observe une pente positive sur la courbe orange). En revanche, dans le cas supervisé, la précision de test se stabilise autour de 66%. Par conséquent, il est raisonnable de conclure que FixMatch améliore l'apprentissage et semble fonctionner de manière satisfaisante.

4.3.2 5% de données labellisées

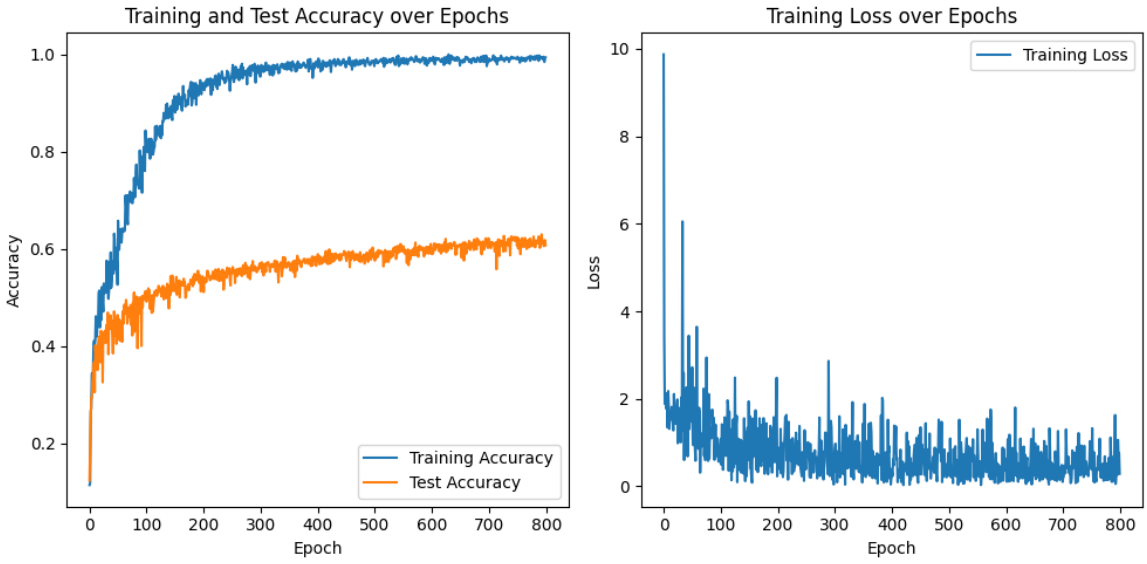


Figure 8: Précisions d'entrainement/de test et la loss au cours des epochs

Dans ce cas, il est observé que la précision de test atteint 63% et est susceptible d'augmenter avec un nombre d'epochs plus élevé (une pente positive est constatée). En revanche, dans le cas supervisé, la précision de test se maintient autour de 52%. Par conséquent, on peut conclure que FixMatch semble améliorer l'apprentissage et semble fonctionner efficacement.

4.3.3 1% de données labellisées

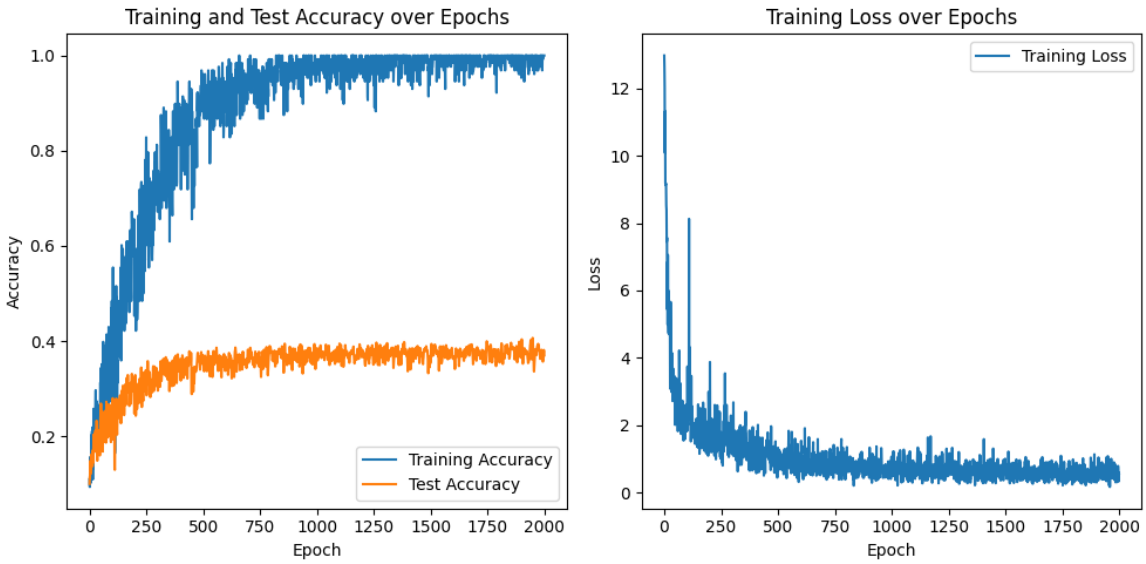


Figure 9: Précisions d'entrainement/de test et la loss au cours des epochs

Dans ce cas, les résultats sont légèrement inférieurs à ceux du cas supervisé, ce qui pourrait être attribué à la sensibilité du modèle au choix initial des données labellisées, comme nous l'expliquerons dans la partie suivante.

5 Choix des meilleures données labellisées

L'utilisation de t-SNE (t-Distributed Stochastic Neighbor Embedding) peut être une approche intéressante pour sélectionner les données labellisées dans le contexte de l'apprentissage semi-supervisé. Avec t-SNE, nous pouvons appliquer l'algorithme aux données non labellisées pour les réduire à une dimension inférieure, ce qui nous permet de visualiser les relations et les clusters dans les données non labellisées.
Pour sélectionner les données labellisées, nous pouvons suivre les étapes suivantes:

1. Appliquer t-SNE : Nous appliquons l'algorithme t-SNE aux données non labellisées pour les réduire à une dimension inférieure, par exemple, 2 ou 3 dimensions. Cela nous permet de visualiser les relations et les regroupements dans les données non labellisées.

2. Visualiser les données non labellisées : Nous traçons les données non labellisées dans l'espace réduit en utilisant t-SNE. Cette visualisation nous permet d'identifier les clusters et les regroupements de données similaires. Cela nous aide à avoir une meilleure compréhension de la structure des données.

3. Échantillonnage stratégique : À partir de la visualisation t-SNE, nous sélectionnons stratégiquement des échantillons dans les clusters qui représentent différentes classes ou régions de l'espace des données. Nous choisissons des échantillons qui semblent être les plus éloignés des autres clusters ou qui sont difficiles à classer visuellement (ceux sur les frontières par exemple). Cela nous permet de sélectionner des échantillons représentatifs.

En utilisant t-SNE pour sélectionner les données labellisées, nous pouvons exploiter la structure des données non labellisées pour choisir stratégiquement les échantillons les plus informatifs à étiqueter. Cela permet d'optimiser l'utilisation des ressources et d'améliorer les performances du modèle d'apprentissage automatique dans un contexte où les données labellisées sont limitées.

Nous avons implémenté un exemple d'entraînement d'un réseau de neurones convolutifs (CNN) sur l'ensemble de données CIFAR-10 et de visualisation des incrustations t-SNE des caractéristiques apprises. Les résultats de la t-SNE figurent dans les deux figures 10 et 11.

Cependant, en raison des limitations liées aux contraintes de capacité de calcul et à la nécessité de réentraîner les modèles avec des batchs de données labellisées issus de t-SNE, nous n'avons pas pu aller plus loin dans notre étude sur l'utilisation de t-SNE pour sélectionner les données labellisées dans le contexte de l'apprentissage semi-supervisé.

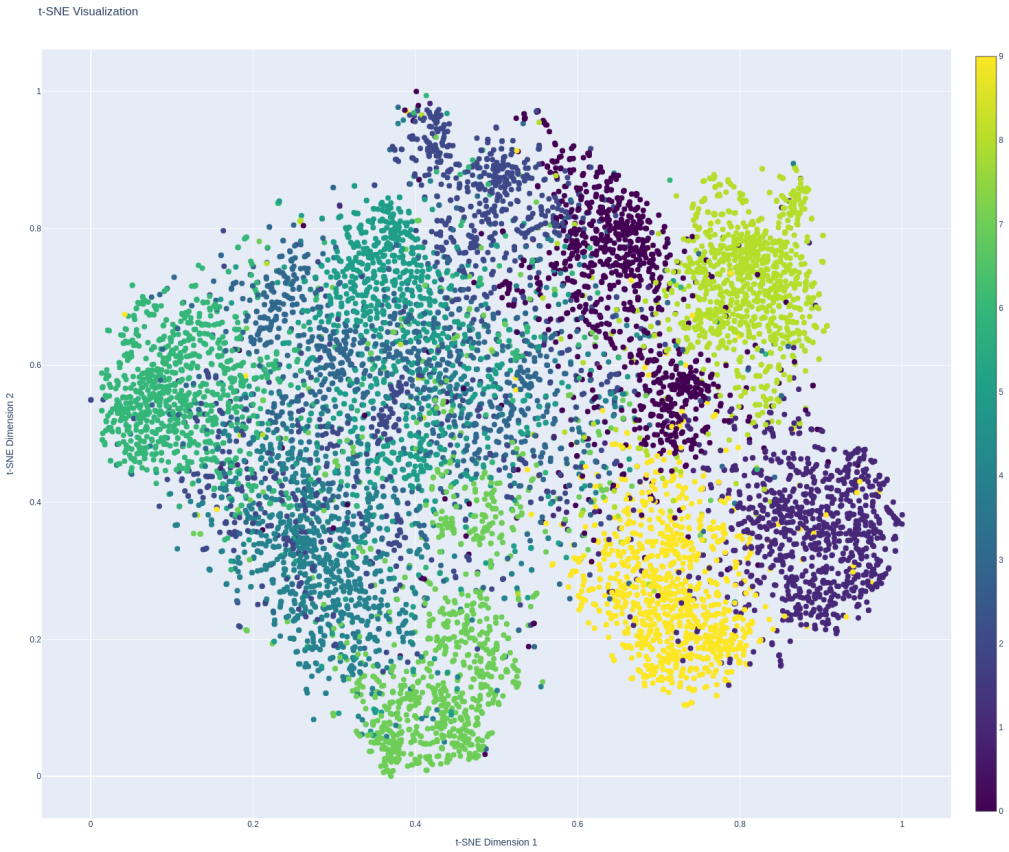


Figure 10: t-SNE sur CIFAR 10

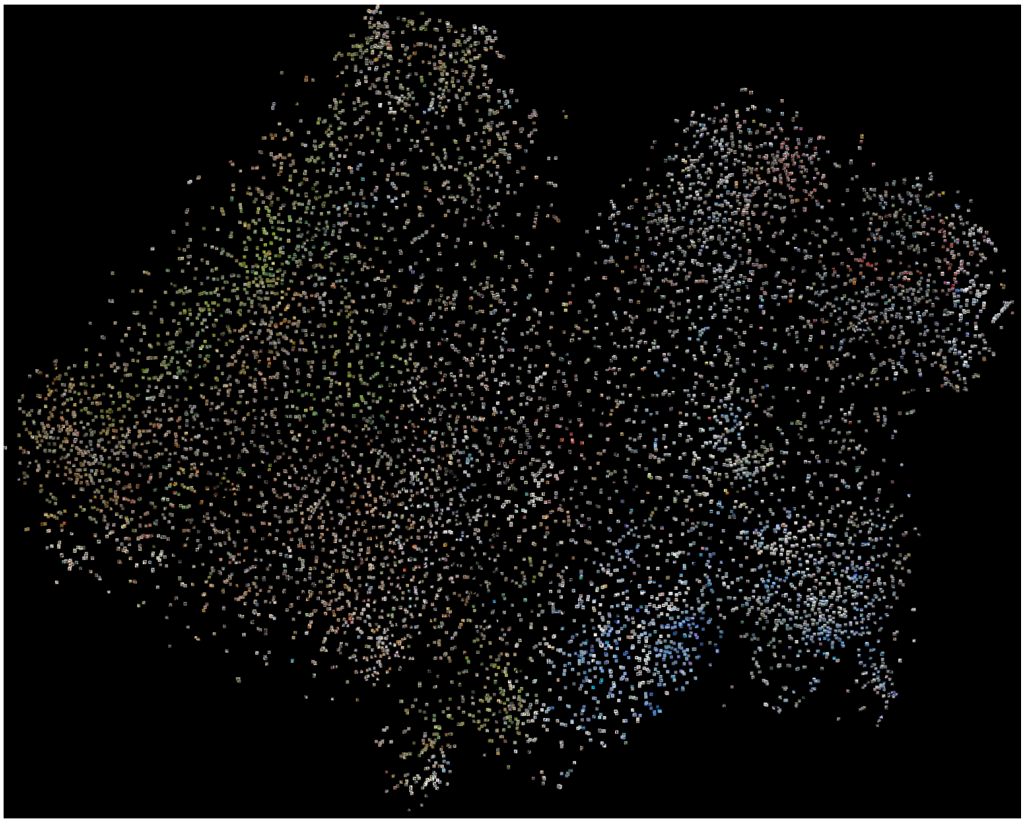


Figure 11: Les images affichées dans la visualisation t-SNE

6 Conclusion

En conclusion, l'algorithme FixMatch s'est révélé efficace pour améliorer les performances des modèles d'apprentissage automatique dans le cas de ressources de données limitées. En utilisant à la fois des données labellisées et non labellisées, et en appliquant des techniques d'augmentation de données et de régularisation de la cohérence, FixMatch a permis d'obtenir de meilleurs résultats par rapport à l'apprentissage supervisé traditionnel.

L'ajout de données non labellisées a permis d'améliorer la capacité de généralisation du modèle, ce qui a conduit à des performances supérieures sur les ensembles de données de test. De plus, l'utilisation de l'augmentation forte RandAugment a contribué à augmenter la robustesse du modèle envers les variations et les déformations des images.

Cependant, il convient de noter que l'apprentissage semi-supervisé peut être sensible au choix initial des données labellisées. Dans le cas de proportions très faibles de données labellisées, les performances peuvent être moins bonnes que celles de l'apprentissage supervisé. Il est donc essentiel de choisir judicieusement les données labellisées pour maximiser l'efficacité de l'apprentissage semi-supervisé.

En conclusion, l'algorithme FixMatch offre une approche prometteuse pour l'apprentissage avec des ressources de données limitées, et mérite d'être exploré plus en profondeur dans des contextes spécifiques où les données labellisées sont rares ou coûteuses à obtenir.

References

[1] Kihyuk Sohn et al. “FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence”. In: (2020). URL: https://arxiv.org/abs/2001.07685.