

Utilisation d'IA générative

Class imbalance :

L'IA générative a été utilisée afin d'avoir des idées de méthodes qui existent pour gérer le déséquilibre dans les classes.

Prompt utilisé :

In the context of a facial expression classification task, the dataset presents a significant class imbalance between emotions.

What methods can be used to handle class imbalance in this type of classification problem?

For each proposed method, please explain the underlying principle and how it can be applied in practice (data-level or model-level).

Résultat :

Le prompt nous a expliqué différentes méthodes :

1. Méthodes au niveau des données

- Oversampling (ex. duplication)
- Undersampling des classes majoritaires
- Data augmentation ciblée (rotations, flips, variations de luminosité)

2. Méthodes au niveau du modèle

- Pondération des classes dans la fonction de perte
- Ajustement des seuils de décision

Regard critique :

Y'avait pas de problème dans les méthodes proposées et l'explication était assez claire et directe, par la suite fallait faire le choix technique et implémenter la méthode dans le code.

Attention Map :

L'IA générative a été utilisée afin d'implémenter la visualisation des attention maps dans les tests du zero-shot avec le modèle CLIP, dans le but d'analyser quelles régions de

l'image influencent la prédiction du modèle (interpréter le comportement du réseau et avoir une explicabilité des résultats).

Prompt utilisé :

```
I am working on a PyTorch project using CLIP and I want to visualize which regions of an input image influence the prediction for a given text prompt. I am looking for an attention map implementation adapted to CLIP's visual encoder.
```

```
Could you explain the steps to generate an attention map and overlay it on the original image for visualization?
```

```
Give me a functional example .
```

Résultat :

Le prompt nous a donné les étapes :

- Hook sur une couche intermédiaire
- Gradients par rapport au texte CLIP
- Pooling spatial des gradients
- Calcul Grad-CAM
- Interpolation vers la taille de l'image
- Visualisation et superposition

Regard critique :

Les sorties générées étaient globalement pertinentes, mais nécessitent des ajustements pour s'adapter à notre code spécifique. Certaines hypothèses implicites (choix de la couche de saliency, format des tenseurs) devaient être vérifiées manuellement (chercher their meaning).

Débogage :

Nous avons également utilisé l'IA générative pour résoudre certains problèmes de débogage, notamment des commandes système comme unzip (pour décompresser les données), la gestion d'erreurs PyTorch ou le traitement de problèmes de dimensions dans les tenseurs. Malheureusement, ces interventions n'ont pas été conservées sous forme de traces détaillées (prompt, résultats..), mais elles ont servi à surmonter rapidement des obstacles techniques ou des recherches d'information light.

Nous avons également utilisé des modèles pour la clarification de certains concepts techniques.