

Rapport Use Case - PRODITEC-1

Equipe:

- **ALAMI** Oumaima
- **BESSAN** Eugene
- **DJE MOINTI** Patrice Jean-Marc O BI

Sommaire

PRODITEC

1. Contexte du Use Case
2. Notre approche
3. Data augmentation
4. Création & Training du modèle
5. Résultats
6. Conclusion & Travaux futurs



Contexte du Use Case

Dans le cadre du contrôle en ligne de comprimés pharmaceutiques basé sur de l'IA, **PRODITEC** a créé un modèle puissant qui produit d'excellents résultats avec une précision de **99,7%**. Or, pour arriver à ce chiffre, le modèle doit être entraîné sur **30 000 images** de pilules dont 15 000 de bonnes pilules et 15 000 de mauvaises pilules réparties en différents types de défauts, ce qui représente un travail colossal pour l'utilisateur afin de labéliser les 30 000 images, sans compter l'inévitable risque d'erreur.

L'objectif de ce cas d'utilisation est d'obtenir le même résultat avec seulement **1000 images** compressées, dont 500 sont bonnes et 500 sont réparties sur différents types de défauts, en utilisant des méthodes d'amélioration des données ou d'autres méthodes.

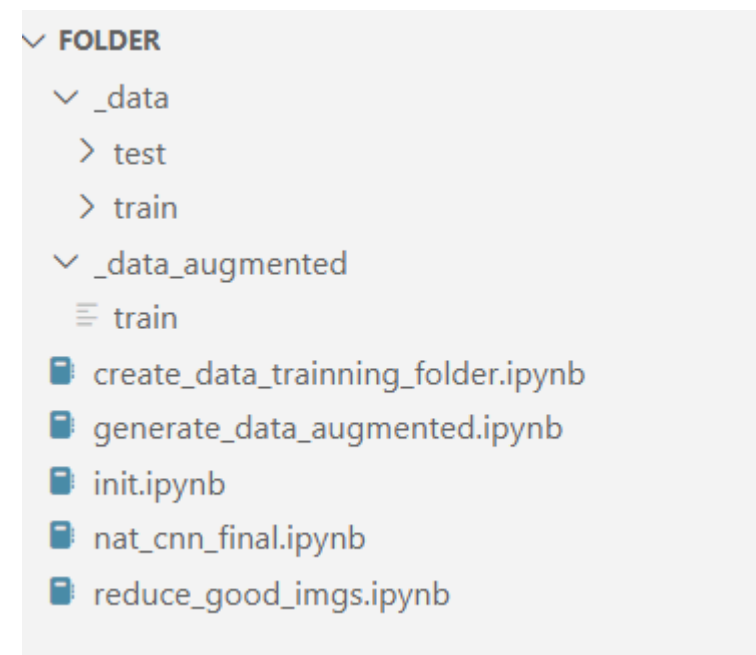
Notre approche

Notre approche à consister effectuer les étapes suivantes :

1. Réorganisation des données de sorte à obtenir 16 classes
2. Pré-augmentation des données
3. Nettoyage des données pré-augmentées
4. Entraînement et évaluation

Pour réaliser cette approche il fallut passer par une séries d'opérations pour aboutir à l'entraînement.

C'est ainsi que nous avons définis l'architecture suivantes

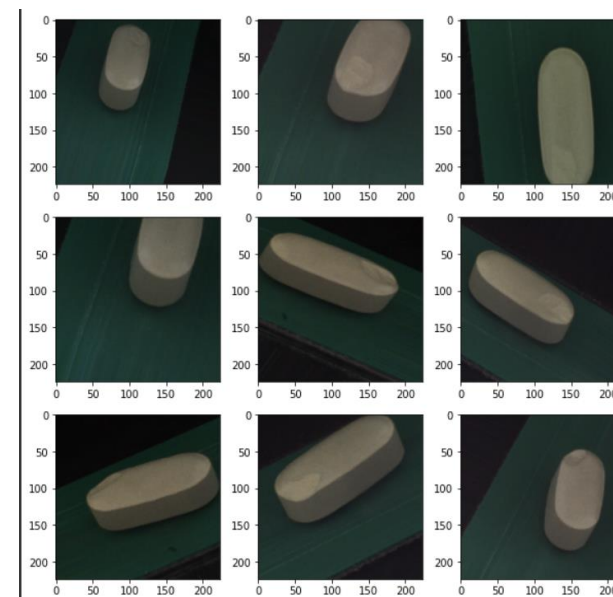


Data augmentation

Compte tenu du manque d'images pour former notre modèle (seulement 1000 images par rapport aux 30000 images utilisées pour la formation de l'approche de Proditec, nous avons décidé d'opter pour l'augmentation des données pour améliorer les performances et les résultats de notre modèle en formant des exemples nouveaux et différents pour entraîner le jeu de données.

L'idée générale est d'appliquer les transformations classiques et d'enregistrer l'ensemble des données dans un dossier pour l'entraînement.

L'avantage est de maîtriser et de contrôler processus génération pour avoir l'ensemble de données le plus adapté pour l'entraînement



Création & Training du modèle

Notre approche consistait à créer un modèle CNN et à l'entraîner. Nous avons importer les bibliothèques requises et charger l'ensemble de données (training dataset et test dataset). Ensuite nous avons appliquer la data augmentation sur les datasets.

Le dictionnaire des labels est créé afin de récupérer les noms de classe par rapport aux indices des labels utilisés pour former le modèle.

Nous avons opter pour le modèle CNN.

L'optimiseur choisi est ADAM

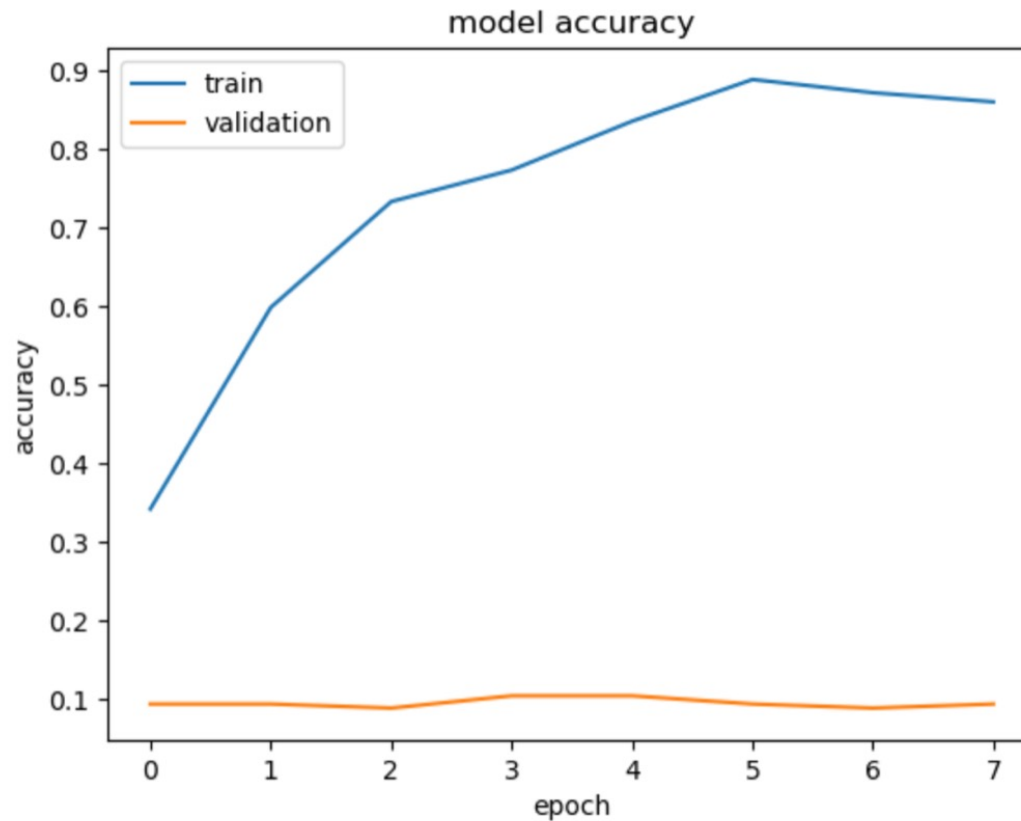
Création & Training du modèle

PRODITEC

Puisque l'ensemble de données d'entraînement est prêt et après la création du modèle CNN, on a entraîné les ensembles de données d'image.

Résultats

PRODITEC



Accuracy: 86%

Val accuracy: 9.36%

Conclusion & Travaux futurs

→ Les modèles sont plus ou moins faciles à mettre en place mais la puissance de calcul nécessaire pour les rejeter ou les retenir rapidement faisait défaut.

Globalement nous retenons que l'usage de **16** classes est une approche qui présente beaucoup d'inconvénients.

Les prochaines étapes:

- **Evaluer** le modèle sur l'ensemble de test à l'aide de différentes métriques de classification.
- **Visualiser** les images présentes dans l'ensemble d'entraînement et de test.

Rapport Use Case - PRODITEC-1

Equipe:

- **ALAMI** Oumaima
- **BESSAN** Eugene
- **DJE MOINTI** Patrice Jean-Marc O BI