Báo cáo

Nhóm gồm 1 thành viên : Phạm Gia Hy | MSSV: 21110311 | Lớp: 21TTH

Nội dung báo cáo:

1. Nội dung cần thiết đều nằm trong tất cả các file notebook, có thể không cần đọc file README này cũng được
2. Trong repo này, ta tiến hành **thí nghiệm với 4 file notebook**.
   - file_1_ml_data_duplicate_hieu_suat_cao
     - Duplicate dữ liệu, với mỗi tấm ảnh, ta nhân bản nó lên 1 lần
     - Tập dữ liệu sau khi duplicate gồm hơn 70.000 images
     - Sử dụng các ML model : **KNN, Gaussian Naive Bayes, Random Forest, MLP** cho dữ liệu **trước và sau PCA** để so sánh hiệu suất (bao gồm độ chính xác và thời gian train)
     - Áp dụng kỹ thuật search siêu tham số
     - Hiệu suất cao nhất đạt 88% khi dùng model : KNN trên tập dữ liệu sau khi đã xử lý PCA
     - (Để biết thêm chi tiết, xem trong notebook)
   - file_2_ml_data_goc
     - Sử dụng dữ liệu gốc
     - Sử dụng 3 model như trong file phía trên, trừ Random Forest, hiệu suất cao nhất đạt được : 41%
   - file_3_dl_**CNN**_52_PERCENT
     - Sử dụng dữ liệu gốc nhưng có thêm **Data Augmentation**
     - Accuracy hội tụ về 52% sau 17 epoch
   - file_4_ml_randomforest_data_goc
     - Sử dụng dữ liệu gốc
     - Hiệu suất được đề cập trong phần 3 bên dưới :
3. Model đạt được hiệu suất cao nhất trên tập dữ liệu gốc : RandomForestClassifier() : 45.4%
   Điểm chung giữa các model khi áp dụng vào data này: **Class 5 lúc nào cũng đạt hiệu suất tốt nhất**
   - **Accuracy (cao nhất khi dùng một ML model) : 45.4%**
   - Class đạt hiệu suất tốt nhất : Class 5
   - Class 0 và Class 4 luôn đạt hiệu suất thấp nhất
   - Class 1 tuy có rất ít dữ liệu so với các Class còn lại, nhưng hiệu suất vẫn ổn định khi cho train đủ lâu
   - **Precision cao nhất cũng là của class 5 (dùng RandomForestClassifier())**
     (với tập dữ liệu đã qua xử lý PCA) : 79%