

AI Engineer Challenge

1. Dataset

Dataset yang digunakan adalah *Construction PPE*. Dataset ini memiliki 16 label yaitu:

['no boots', 'Safety Boot', 'hat', 'gloves', 'Human', 'no vest', 'no hat', 'no boot', 'no gloves', 'helmet', 'Safety Vest', 'vest', 'Helmet', 'Gloves', 'boots', 'glasses'].

Dataset ini juga memiliki beragam gambar dari berbagai tempat serta pencahayaan yang baik, juga dataset ini memiliki beberapa angle pengambilan gambar.

2. Penggunaan Algoritma

Algoritma yang digunakan adalah Faster-RCNN. Algoritma ini menawarkan kemampuan deteksi objek yang efisien dan andal berkat arsitektur Convolutional Neural Network (CNN) yang canggih dan penggunaan region proposal network (RPN) untuk mengidentifikasi kandidat-kandidat objek. Dengan kecepatan yang cukup tinggi dan kemampuan untuk mendeteksi objek dengan akurasi yang tinggi, Faster R-CNN cocok untuk berbagai aplikasi deteksi objek yang memerlukan respons cepat dan tingkat keandalan yang tinggi.

Model yang digunakan yaitu Resnet50. Menggunakan pre-trained model ResNet-50 menjadi pilihan yang bijaksana dalam banyak kasus karena menggabungkan manfaat transfer learning, kinerja yang terbukti, dan efisiensi sumber daya. Model ini telah dilatih dengan dataset besar seperti ImageNet, sehingga memiliki kemampuan untuk mengekstraksi fitur-fitur umum dari gambar. Dengan menggunakan pre-trained model ini, pengguna dapat mengharapkan kinerja yang baik dalam tugas-tugas seperti klasifikasi gambar dan deteksi objek tanpa perlu melatih model dari awal.

3. Evaluasi Kinerja Model

Model di train dengan jumlah epoch sebesar 25 dan nilai Loss sebesar 0.488. Ini menandakan bahwa model belum optimal dalam mempelajari dataset. Hal ini bisa disebabkan oleh berbagai faktor, salah satunya adalah overfitting, di mana model terlalu kompleks dan "menghafal" data pelatihan sehingga tidak mampu menggeneralisasi dengan baik pada data test.

Accuracy dari model adalah 0.2195 dan Precision adalah NaN.

```
accuracy = np.trace(conf_matrix) / np.sum(conf_matrix)
print(accuracy)

0.21954887218045113

precision = np.diag(conf_matrix) / np.sum(conf_matrix, axis=0)
average_precision = np.mean(precision)
print(average_precision)

nan
```