**ABSTRACT:**

When we talk about the Pre-trained model in the Computer Vision domain, Alexnet comes out as a leading architecture.

We have tried to use layers similar to the AlexNET to meet our problem i.e. to classify scenarios. To accelerate the process we have used the TPUs. The reports of the model have been displayed using weights and biases.

**Key points:**

* Used **TPUs** to accelerate the alexnet model training due to usage of large number of data.
* Report generation using **Weights and Biases -** Weights & Biases is the machine learning platform for developers to build better models faster. Use W&B's lightweight, interoperable tools to quickly track experiments, version and iterate on datasets, evaluate model performance, reproduce models, visualize results and spot regressions, and share findings with colleagues.

**DATASET:**

The dataset we have used is the places365.

Shape of image : (256,256,3)

We have **3,00,000 images in the train dataset classified into 50 classes.**

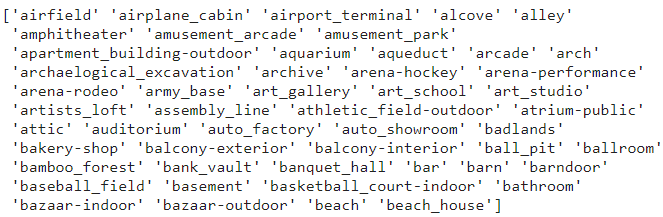
Pre-processing steps include

* Scaling pixel values within the range of 0-1 (divided by 255)
* Mean normalisation on all images

**ENVIRONMENT:**

The model is trained on **Kaggle Notebooks** environment using **TPU VM V3-8** accelerator to speed up the training process.

Sample labels in the dataset:

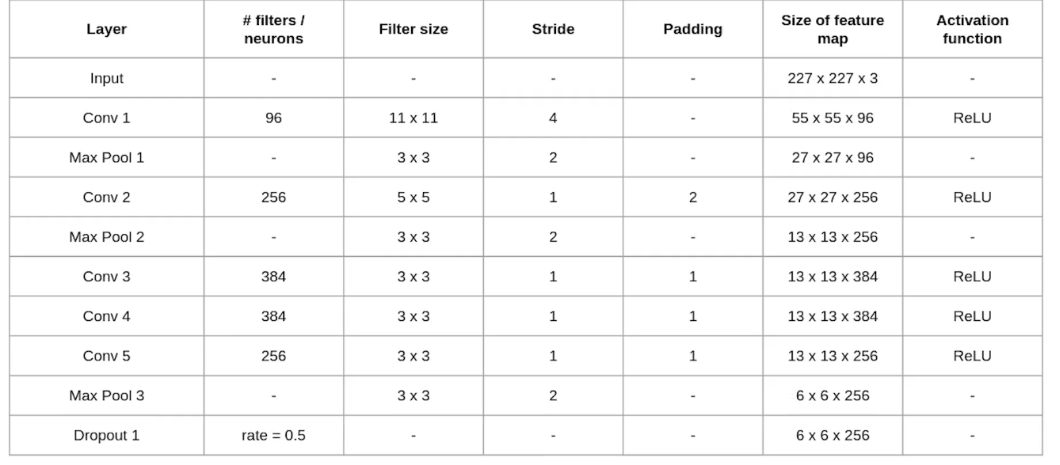


Sample images in the train dataset:



**Variant of AlexNET MODEL:**

**Architecture:**

****

**Compilation:**

Model is compiled using Adam optimiser and categorical cross entropy loss function.

Due to the large variety of scenarios dealt by the model, it would be apt to use **top\_k\_categorical\_accuracy** wherein we consider the top 5 probable label as true positive rather than 1.

**Training:**

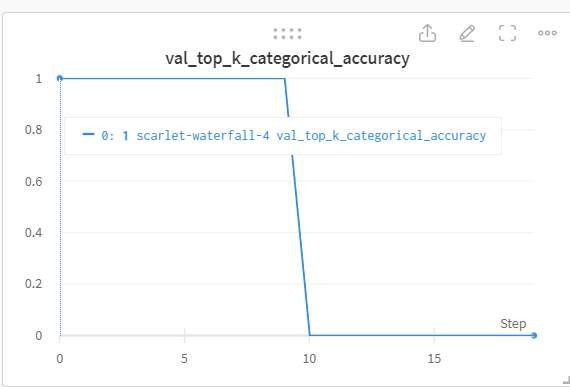
Model is trained with a batch size of 32 (7000 batches for 20 epochs).

Results obtained after 20 epochs:

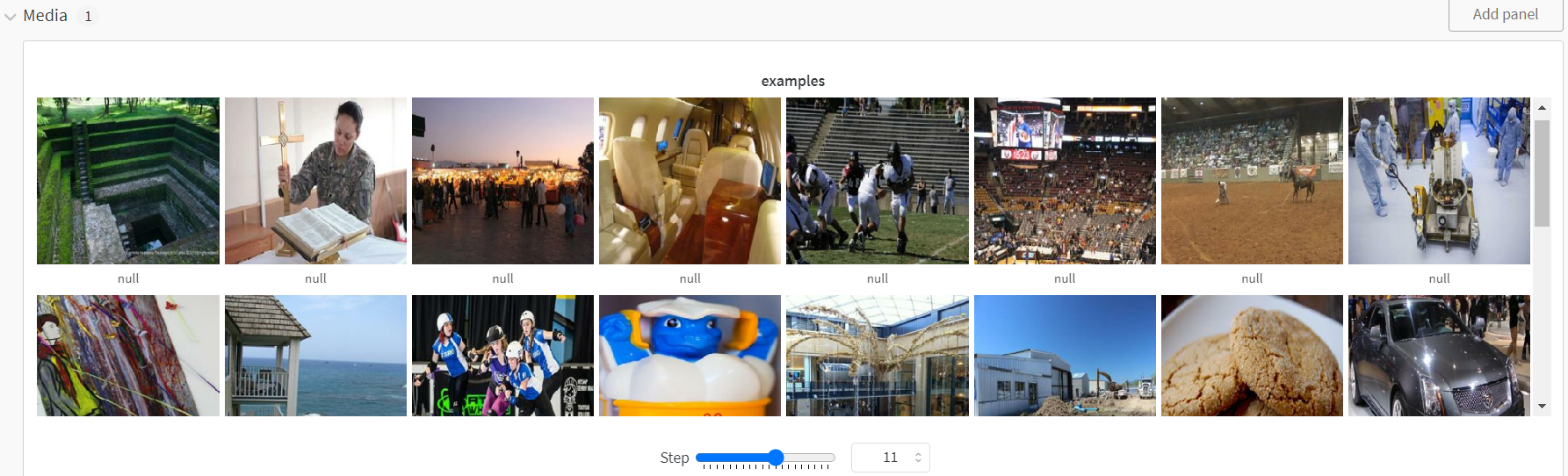
* **Train accuracy : 96.63%**
* **Validation accuracy : 95.36%**
* **Testing : 75%**

Reports from **Weights and Biases**





Test data:



GPU AND TPU power used:





