Image Classification with Convolutional Neural Networks

This document summarises the process of using a Convolutional Neural Network (CNN) to classify images.

Model Choice:

- CNNs are ideal for image classification tasks like satellite imagery, medical imaging, time series forecasting, and anomaly detection.
- They excel at extracting features from data through multiple layers.

Key Components:

1. Convolution Layers:
   - The core building block, involving a filter and an input image.
   - The filter (kernel) scans the image for specific features, performing a dot product at each location.
   - This process, called convolution, repeats until the entire image is covered.
2. Activation Functions:
   - Introduce non-linearity into the network, allowing it to learn complex patterns.
3. Pooling Layers:
   - Reduce the number of parameters and spatial dimensions of the data.
   - Max pooling takes the maximum value within a window, while average pooling calculates the average.

Model Architecture:

- A sequential CNN model is used, consisting of:
  - Convolutional layer: With 32 filters, 3x3 kernel, and ReLU activation for feature extraction.
  - Max pooling layer: Reduces the data size by half.
  - Flatten layer: Prepares the data for the fully-connected layers.
  - Dense layers: With 256 and 512 units and ReLU activation for learning complex relationships.
  - Dropout layer: Prevents overfitting by randomly dropping units during training.
  - Output layer: With 5 units and softmax activation for multi-class classification.

Training Process:

- Image Preprocessing:
    - Loads and converts images to RGB, resizes them, and converts them to NumPy arrays.
- Model Training:
    - Data is split 70/30 for training and testing.
    - Pixel values are normalized between 0 and 1.
    - Model is trained for 200 epochs with a 10% validation split.
- Model Compilation:
    - Utilizes the Adam optimizer, sparse_categorical_crossentropy loss, and accuracy metric.

Critical Findings:

Although the model has achieved an admirable accuracy of 83% after 160 epochs, its performance is likely constrained by the dataset's limited size, comprising only 150 images. This small dataset poses a challenge for the model to grasp generalized features, potentially resulting in overfitting tendencies. To enhance the model's performance, it would be beneficial to augment the training dataset by increasing its size and diversity. Additionally, experimenting with various regularization techniques can be explored as a means to mitigate overfitting.

Overall:

This approach demonstrates how CNNs leverage their strengths in feature extraction and spatial awareness for image classification tasks.