# Applied Computer Vision Assignment 3

## Carnegie Mellon University, Africa

JANUARY - MAY 2024

**OLATUNJI Damilare Emmanuel**
dolatunj@andrew.cmu.edu
Engineering Artificial Intelligence | Class of 2025

**INSTRUCTORS:**

| | |
|---|---|
| Moise Busogi: | mbusogi@andrew.cmu.edu |
| Assane Gueye | assaneg@andrew.cmu.edu |

**TEACHING ASSISTANTS**

## Question 1

1. Why does the CNN architecture perform better than the MLP architecture? (Hint: The CNN architecture does not begin with a flattened layer like the MLP. Other useful answers are acceptable as well.).

**Answer:**

a. MLPs requires a flattened layer as their first layer, which loses spatial information present in the input data. This loss makes MLPs less effective at capturing local patterns and features, resulting in poorer performance compared to CNNs.

b. CNNs share weights across different regions of the input data, meaning that the same filter is applied to different parts of the image. This measure reduces the number of trainable parameters compared to MLPs, making CNNs less prone to overfitting and more computationally efficient.

c. CNNs are typically more parameter-efficient compared to MLPs, especially for tasks involving large input data such as images. By sharing weights and exploiting local connectivity, CNNs require fewer parameters to learn complex features from high-dimensional input data, making them more scalable and easier to train on large datasets.

d. CNNs exploit the spatial locality of data through convolutional layers. Each neuron in a convolutional layer is connected only to a local region of the input data, allowing the network to learn spatial hierarchies of features. Moreover, CNNs utilize shared weights (or kernels) across the input, reducing the number of parameters and enabling the network to learn translational invariance. This local connectivity and weight sharing enable CNNs to effectively capture spatial patterns in the data, which is crucial for tasks like image recognition.

2. In terms of updating the weights of the CNN filter, why is a LeakyReLU activation function preferable to a ReLU activation function?

**Answer:**

a. The main advantage of LeakyReLU over ReLU is its ability to mitigate the "dying ReLU" problem. In ReLU, neurons can sometimes become inactive (or "dead") for certain inputs, leading to zero gradients during backpropagation and halting the learning process. LeakyReLU introduces a small slope (typically a small negative slope) for negative inputs, ensuring that even neurons that are not activated by positive inputs still receive a small gradient during backpropagation. This helps to prevent neurons from becoming completely inactive and facilitates continuous learning.

b. LeakyReLU can contribute to more stable training dynamics, especially in situations where ReLU might suffer from dead neurons or vanishing gradients. This stability can lead to faster convergence and better overall performance during training.

3. Explain the use of Batch Normalization in a CNN architecture? Secondly, does Batch Normalization prevent overfitting? If yes or no, why?

a. Batch Normalization (BN) is a pivotal technique in Convolutional Neural Network (CNN) architectures, primarily aimed at enhancing training speed, stability, and model performance. It functions by standardizing the activations of each layer across mini batches during training. It also helps to address the issue of internal covariate shift, which can occur when the distribution of the inputs changes during training, making the learning process more difficult.

b. Yes, batch normalization helps prevent overfitting.
   Batch Normalization does not directly prevent overfitting, but it can indirectly contribute to mitigating it to some extent. While the primary purpose of Batch Normalization is to stabilize training and improve convergence speed by addressing internal covariate shift, its stochastic nature during training introduces noise to the learning process. This noise acts as a form of regularization, similar to techniques like dropout, by adding variability to the activations across mini-batches.
   Batch Normalization alone may not suffice to entirely prevent overfitting. Effective prevention of overfitting typically requires a combination of regularization techniques, appropriate model complexity control, and sufficient training data. Therefore, while Batch Normalization contributes to regularization, its effectiveness in combating overfitting depends on various factors and should be complemented with other strategies for optimal results.

4. The CNN architecture in Figure 2 incorporates a Dropout of 0.5, what is the impact of this and how does it affect the neurons in the architecture?

**Answer:**

- The dropout layer with a rate of 0.5 in the CNN architecture of Figure 2 means that during training, 50% of the neurons in the layer will be randomly dropped out. This has the following effects:

a. **Reduces overfitting:** By randomly dropping out neurons, the network is forced to learn redundant features and avoid relying on any one neuron too heavily. This helps to improve the generalization of the model, meaning it will perform better on unseen data.
b. **Prevents co-adaptation:** When neurons are always working together, they can become co-adapted, meaning they rely on each other to make predictions. Dropout helps to prevent this by forcing the network to learn independently of any specific neuron.
c. **Increases robustness:** By dropping out neurons, the network becomes more robust to noise and variations in the data. This can help to improve the overall performance of the model.

**PART II**

1. Document the model architecture, including the choice of layers, activation functions and any regularization techniques used.

**Model Architecture**

For this assignment, the custom-CNN architecture consists of convolutional layers for feature extraction followed by fully connected layers for classification. ReLU activation functions are used throughout the network, and the softmax activation function is used in the output layer for multi-class classification. The table below explains in detail.

| | |
|---|---|
| **Input Layer** | Resized and rescaled image. |
| **Convolutional Layers** | 1st Convolutional Layer:<br>- 32 filters with a kernel size of (3, 3) and ReLU activation function.<br>- MaxPooling layer with a pool size of (2, 2) to downsample the feature maps.<br>2nd Convolutional Layer:<br>- 64 filters with a kernel size of (3, 3) and ReLU activation function.<br>- MaxPooling layer with a pool size of (2, 2) to downsample the feature maps.<br>3rd Convolutional Layer:<br>- 64 filters with a kernel size of (3, 3) and ReLU activation function.<br>- MaxPooling layer with a pool size of (2, 2) to downsample the feature maps.<br>4th Convolutional Layer:<br>- 64 filters with a kernel size of (3, 3) and ReLU activation function.<br>- MaxPooling layer with a pool size of (2, 2) to downsample the feature maps.<br>5th Convolutional Layer:<br>- 64 filters with a kernel size of (3, 3) and ReLU activation function.<br>- MaxPooling layer with a pool size of (2, 2) to downsample the feature maps. |
| **Flatten Layer** | Outputs of the convolutional layers are flattened to prepare for the fully connected layers |
| **Fully Connected Layers** | 1 Dense layer with 64 units and ReLU activation function |
| **Output layer** | Dense layer with “num_classes” units and softmax activation function for multi-classification. |

**Regularization Technique:** No explicit regularization techniques such dropout or L2 regularization was used.

## Custom CNN Performance EvaluationTraining Dataset

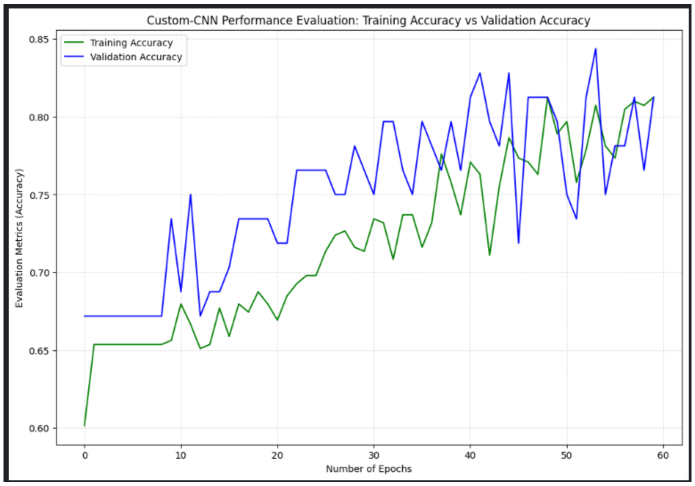


**Fig. 1.1a Custom-CNN Training & Validation Accuracy**

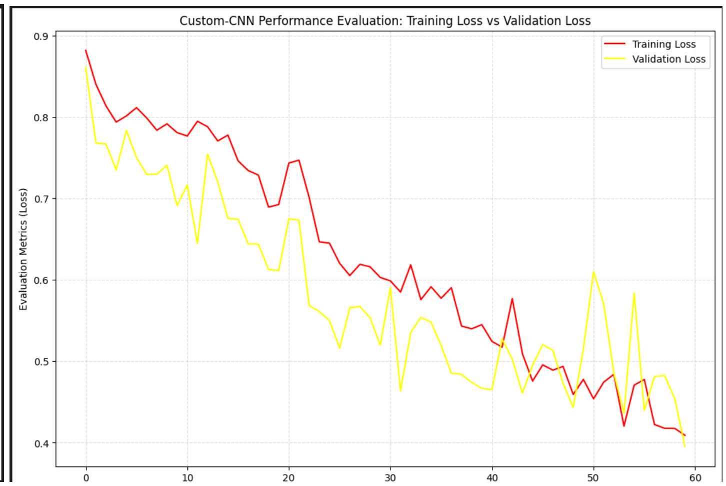


**Fig. 1.1b Custom-CNN Training & Validation Loss**

2. Evaluate the performance of all the models on the test set using appropriate metrics.
   The selected neural networks used for comparison includes Alex-Net, VGG-16 and VGG-19.

| Metric | Custom-CNN | Alex-Net | VGG-16 Net |
|---|---|---|---|
| Training Accuracy | 0.8229 (82%) | 0.6354 (64%) | 0.6146 (61%) |

Table 3.1 Comparison of Training Accuracy – Custom CNN, Alex-Net and VGG-16 Net

The performance of each of the model could be analysed using confusion matrix techniques.

**Parameter Definition:**

- True Positive (TP): The number of cases where the model correctly predicted the positive class.
- True Negative (TN): The number of cases where the model correctly predicted the negative class.
- False Positive (FP): The number of cases where the model incorrectly predicted the positive class.
- False Negative (FN): The number of cases where the model incorrectly predicted the negative class.


**Fig. 3.1 Custom-CNN Implementation on Test Data**

| Category | No. of Occurrences | TP | TN | FP | FN |
|---|---|---|---|---|---|
| Full water level | 7 | 3 | 0 | 0 | 0 |
| Half water level | 1 | 2 | 0 | 0 | 0 |
| Overflowing | 1 | 0 | 0 | 0 | 0 |

**Explanation:** With seven (7) instances of the full water level category and two (2) instances of half water level, the custom-CNN model correctly identified all occurrences, demonstrating its high accuracy and reliability in detecting full water level. Additionally, there were no false positives or negatives, indicating precise prediction across board.


**Fig. 3.1 Alex-Net Implementation on Test Data**

## Explanation:

With seven (7) instance of the full water level category, two (2) instance of the half water level, and one (1) of the overflowing water level categories, the Alex-Net model only predicted three (3) instances correctly for the full water level category and wrongly predicted three (3) instance of full water level category for overflowing. From the two (2) instances of half water level category, one was rightly predicted and the other confused for overflowing. Lastly the overflowing was rightly predicted.


**Fig. 3.1 VGG-16 Implementation on Test Data**

## Explanation:

With eight (8) instances of the full water level category and one (1) instance of half water level, the VGG-16 model correctly identified all the full water category occurrences, demonstrating its high accuracy and reliability in detecting full water level. However, it wrongly predicted for half water category as full water.

3. Compare the performance of your custom CNN model with the selected known architectures based on the evaluation metrics. Provide a detailed analysis of the result, discussing the strengths and weaknesses of each model in the context of this classification task.

**Answer:**

- The custom CNN model demonstrates a notably higher training accuracy of 82%, showcasing its ability to effectively learn and generalize from the training data. This indicates that the model has successfully captured the underlying patterns and features present in the dataset, leading to robust predictions.
- Despite its superior training accuracy, the custom CNN model maintains comparable performance to the VGG-16 architecture on the test data. It accurately identifies all instances of the full water level category, highlighting its reliability in real-world scenarios.
- Also, the evaluation reveals shortcomings in both the AlexNet and VGG-16 architectures, particularly in accurately predicting instances of overflowing and half water levels. This underscores the custom CNN's advantage in effectively distinguishing between different water level categories, showcasing its robustness in classification tasks.
- Lastly, the custom CNN model's design offers advantages in terms of computational efficiency and resource utilization compared to larger architectures like VGG-16. For this deployment of such a project like this, custom CNN would be the best choice for the classification problem. This in turn makes it a practical choice for deployment applications.

4. Discuss at least 3 potential applications of the developed model in real-world scenarios.
   a. **Satellite Imagery Analysis:** CNNs are well-suited for analysing satellite imagery to identify patterns, objects, and land features. They can be used for tasks like land-use classification, object detection (e.g., buildings, cars, ships), and change detection (e.g., deforestation, urbanization).
   b. **Autonomous Vehicles:** Convolutional neural networks, including AlexNet, VGG-16, VGG-19, and custom CNN models, are crucial for the perception system of autonomous vehicles. They can be used to process images from onboard cameras and other sensors to detect and classify objects like traffic signs, pedestrians, and other vehicles.
   **c.** **Medical Imaging:** AlexNet, VGG-16, and VGG-19 architectures, can be applied to medical imaging tasks like lesion detection, tumor segmentation, and disease diagnosis. These models can analyze medical images, such as X-rays, MRIs, or CT scans, to identify patterns and anomalies that may indicate specific medical conditions.
   **d.** Text Recognition in Images: CNNs can be used for identifying and classifying individual characters within an image. This is particularly useful in applications like license plat recognition, document digitization and many others.

## BIBILOGRAPHIES

1. “What are the advantages of CNNs over MLPs if a CNN can be represented as a MLP and an MLP can be trained faster (matrix multiplication is...,” Quora, 2019. https://www.quora.com/What-are-the-advantages-of-CNNs-over-MLPs-if-a-CNN-can-be-represented-as-a-MLP-and-an-MLP-can-be-trained-faster-matrix-multiplication-is-faster-than-convolution (accessed Mar. 09, 2024).
2. J. Brownlee, “A Gentle Introduction to the Rectified Linear Unit (ReLU) for Deep Learning Neural Networks,” Machine Learning Mastery, Apr. 20, 2019. https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/
3. “Would you use batch normalization? If so, can you explain why?,” Quora, 2019. https://www.quora.com/Would-you-use-batch-normalization-If-so-can-you-explain-why (accessed Mar. 09, 2024).
4.