**Internship Report  
  
Visualization of Activation Maps for Emotion Detection**

**By: Chetana J Pawar**

**CONTENTS**

**1. INTRODUCTION**

**2. BACKGROUND**

**3. LEARNING OBJECTIVES**

**4. ACTIVITIES AND TASKS**

**5. SKILLS AND COMPETENCIES**

**6. FEEDBACK AND EVIDENCE**

**7. CHALLENGES AND SOLUTIONS**

**8. OUTCOMES AND IMPACT**

**9. CONCLUSION**

**1. Introduction:**

This report documents the process of visualizing activation maps in a convolutional neural network (CNN) for classifying pebbles and shells. The task involved using a pre-trained model to understand which image regions activate CNN filters, providing insights into model interpretability.

Image classification using deep learning has gained significant attention due to its applications in fields such as environmental monitoring, marine biology, and automated sorting systems. However, understanding how CNNs make predictions remains a challenge, as these models function as black boxes. Activation maps help bridge this gap by providing a visual representation of which parts of an image contribute most to the model’s decision-making process. By interpreting these activations, we can assess the reliability of the model, detect biases, and refine its accuracy.

In this report, we explore different pre-trained models, such as VGG16, ResNet50, and Xception, to classify the images and then generate activation maps for the images. We also discuss the techniques employed for visualization and the significance of these insights in improving model transparency and performance.

**2. Background:**

Deep learning models, particularly CNNs, are widely used in image classification tasks, including object recognition and natural element classification. Activation maps highlight the regions that contribute most to the model’s predictions, helping in model debugging and transparency. The primary goal was to leverage a pre-trained model to visualize these activations effectively.

The model used in this task utilizes multiple convolutional layers, pooling layers, and fully connected layers to classify images into pebble and shell categories. The convolutional layers detect spatial features such as edges and textures, while the pooling layers help in reducing dimensionality and computational complexity. The final layers aggregate learned features to produce the classification output.

Pre-trained convolutional neural networks (CNNs) like VGG16, ResNet50, and Xception have been instrumental in advancing computer vision tasks, including object classification. These models, trained on large datasets such as ImageNet, can be fine-tuned for specific applications, leveraging their learned feature representations.

Understanding the internal workings of these pre-trained models, particularly their convolutional layers, is crucial for tasks like visualizing activation maps. These layers are responsible for detecting various features such as edges, textures, and complex patterns, which are essential in distinguishing between pebbles and shells in images.

**3. Learning Objectives**

* Understand the role of CNN activation maps in pebble and shell classification and how they contribute to model interpretability.
* Implement visualization techniques to highlight key regions in images that influence model predictions.
* Develop an understanding of convolutional layers and their role in detecting various features like edges, textures, and shapes.
* Gain hands-on experience in model debugging by analysing activation maps and identifying potential biases or weaknesses in classification.
* Enhance skills in handling image datasets, preprocessing images, and applying neural network interpretability techniques.
* Learn to critically evaluate deep learning models by interpreting their decision-making process through visual insights.

**4. Activities and Tasks**

* Loaded image dataset and pre-processed it for CNN input.
* Selected a pre-trained model suitable for feature extraction.
* Extracted activation maps and visualized them using techniques like Grad-CAM.
* Analysed model behaviour by comparing different layers’ activations.

**5. Skills and Competencies**

* Proficiency in TensorFlow and Keras for deep learning tasks.
* Experience with OpenCV and Matplotlib for image processing and visualization.
* Understanding of neural network interpretability and feature extraction.
* Ability to handle large datasets and optimize model performance.

**6. Feedback and Evidence**

* Successfully visualized activation maps for multiple images of pebbles and shells.
* Identified key image regions that influenced model predictions.
* Compared activation maps across different architectures to assess effectiveness.

**7. Challenges and Solutions**

* **Challenge:** Selecting the appropriate layer for visualization.
  + **Solution:** Experimented with different convolutional layers to identify the most informative one. Layers closer to the output provided more class-discriminative features, while early layers captured more general patterns.
* **Challenge:** Managing computational efficiency.
  + **Solution:** Used pre-trained models with optimized settings to balance accuracy and speed. Additionally, batch processing techniques and GPU acceleration were leveraged to improve computational efficiency.
* **Challenge:** Interpreting activation maps effectively.
  + **Solution:** Cross-referenced multiple visualization techniques, including Grad-CAM and feature occlusion methods, to ensure a comprehensive understanding of model decisions.

**8. Outcomes and Impact**

* Improved understanding of CNN interpretability techniques.
* Gained hands-on experience in model debugging and analysis.
* Developed a systematic approach for evaluating model predictions.

**9. Conclusion**

This internship task provided valuable insights into CNN interpretability through activation map visualization. By leveraging pre-trained models and visualization techniques, the task enhanced our understanding of how deep learning models perceive and classify pebbles and shells in images. Future work may involve integrating these insights to improve model accuracy and robustness.