**Image Classification with Transfer Learning**

**Name: Sairithik singamalla**

**ID: 811286106**

**1. Summary**

This project focuses on classifying images into multiple categories using deep learning techniques. I worked with the **Caltech 101 dataset**, which includes 102 unique categories, ranging from objects like airplanes to animals like wild cats. Leveraging the **MobileNetV2 model**, pretrained on the **ImageNet dataset**, I was able to achieve an impressive validation accuracy of **90.38%**. To optimize the model, I implemented techniques like **early stopping** and **performance monitoring**, ensuring the training was efficient and accurate.

**2. Introduction**

Image classification is one of the most impactful applications of deep learning, finding uses in industries like healthcare, autonomous driving, and retail. The goal of this project was to:

1. Train a robust model capable of classifying images into 102 diverse categories.
2. Utilize transfer learning to reduce training time while maintaining high accuracy.
3. Apply best practices in deep learning to monitor, optimize, and validate the model's performance.

By using the **Caltech 101 dataset**, known for its variety and compact size, I explored how transfer learning can simplify complex tasks. This dataset provided a practical challenge of categorizing diverse images with limited data.

**3. Current Research**

The project builds upon several key advancements in image classification and transfer learning:

* **Transfer Learning**: Using pretrained models like MobileNetV2, ResNet, and EfficientNet has revolutionized deep learning by significantly reducing the computational resources needed for training from scratch.
* **Data Augmentation**: Methods like flipping, rotating, zooming, and cropping images improve model robustness by effectively expanding the dataset.
* **Optimization Tools**: Researchers often use tools like Tensor Board to track metrics, visualize loss curves, and optimize training in real-time.

These techniques are widely adopted in academic and industrial applications, forming the foundation for this project.

**4. Data Collection and Model Development**

**Dataset**

I used the Caltech 101 dataset, which contains:

* 7,356 images for training.
* 1,788 images for validation.
* A total of 102 categories, including examples like "airplanes," "wild cats," "motorbikes," and "bonsai."

**Preprocessing**

To prepare the data for training, I performed the following steps:

1. **Resizing**: All images were resized to **224x224 pixels**, matching the input size required by MobileNetV2.
2. **Normalization**: Pixel values were scaled to the range [0, 1] to speed up convergence.
3. **Data Splitting**: The dataset was split into **80% training** and **20% validation** subsets to evaluate performance during training.

**Model Development**

I selected **MobileNetV2**, a lightweight yet powerful deep learning model, pretrained on the **ImageNet dataset**. Here’s how the model was customized:

* **Base Model**: The pretrained layers of MobileNetV2 were frozen to retain its feature extraction capabilities.
* **Custom Layers**: I added a global average pooling layer, followed by fully connected dense layers and a SoftMax output layer for the 102 categories.
* **Optimizer**: The **Adam optimizer** was used with a learning rate of **0.001** to balance learning speed and stability.
* **Loss Function**: **Categorical cross-entropy** was chosen, as it’s ideal for multi-class classification problems.

**Training**

The model was trained for 10 epochs, and I used early stopping to avoid overfitting. Performance was monitored closely using validation accuracy and loss.

**5. Results**

The training and validation performance across epochs is summarized in the table below:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Epoch** | |  | | --- | | **Training Accuracy** |  |  | | --- | |  | | **Validation Accuracy** | |  | | --- | |  |  |  | | --- | | **Training Loss** | | **Validation Loss** |
| 1 | 62.58% | 84.49% | 1.8497 | 0.5565 |
| 2 | 87.50% | 84.26% | 0.5727 | 0.5570 |
| 3 | 95.00% | 89.15% | 0.1749 | 0.3974 |
| 4 | 96.88% | 88.86% | 0.0959 | 0.4123 |
| 5 | 97.84% | 87.73% | 0.0744 | 0.4409 |
| 6 | 100.00% | 87.67% | 0.0472 | 0.4513 |
| 7 | 98.95% | 88.69% | 0.0407 | 0.3940 |
| 8 | 100.00% | 88.75% | 0.0430 | 0.3971 |
| 9 | 99.59% | 90.11% | 0.0202 | 0.3654 |
| 10 | 96.88% | 90.28% | 0.0335 | 0.3649 |

**Predicted Example**

One of the test images, labelled as a "wild cat," was correctly predicted by the model. This example highlights the model’s ability to generalize across diverse categories, even with limited data.

**6. Observations**

* Accuracy: The validation accuracy peaked at 90.38%, indicating the model successfully learned from the dataset.
* Efficiency: By leveraging transfer learning, I significantly reduced training time and computational requirements compared to training from scratch.
* Challenges: Certain categories, such as "airplanes" and "helicopters," were more challenging to classify due to visual similarities between them.

**7. References**

1**.** Li, F.-F., Andreeto, M., Ranzato, M., & Perona, P. (2022). *Caltech 101 (1.0)* [Data set]. CaltechDATA. <https://data.caltech.edu/records/mzrjq-6wc02>

2. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4510–4520. <https://doi.org/10.1109/CVPR.2018.00474>

3. TensorFlow/Keras Documentation. <https://www.tensorflow.org>

**8. Conclusion**

This project demonstrated the effectiveness of transfer learning for image classification tasks. By using a pretrained model like MobileNetV2, I achieved high accuracy with minimal training effort.

**Key Takeaways:**

1. **Preprocessing**: Ensuring data is correctly pre-processed is essential for smooth and effective training.
2. **Performance Monitoring**: Regularly tracking metrics like validation accuracy and loss helps prevent overfitting and ensures consistent performance.
3. **Transfer Learning**: Leveraging pretrained models drastically reduces the effort and resources needed to build high-performing models.

Overall, this project deepened my understanding of transfer learning and its practical applications in image classification, paving the way for future exploration of similar tasks in other domains.