DermAI

Our approach to developing an image classification model to differentiate between bruises, melanoma, and nevus from medical images involved several structured steps:

1. Data Acquisition and Pre-processing:

The dataset, sourced from Roboflow, each class was a dataset alone then, we gathered each dataset and created a class from each and split it into a training and testing split, and then we put it in a ZIP file containing images of {bruises, melanoma, and nevus}, was programmatically extracted into a designated directory for processing. The pre-processing steps included:

Applying transformations to augment the training data, which included random resizing, cropping, horizontal flipping, and rotation to introduce variability and improve the model's ability to generalize.

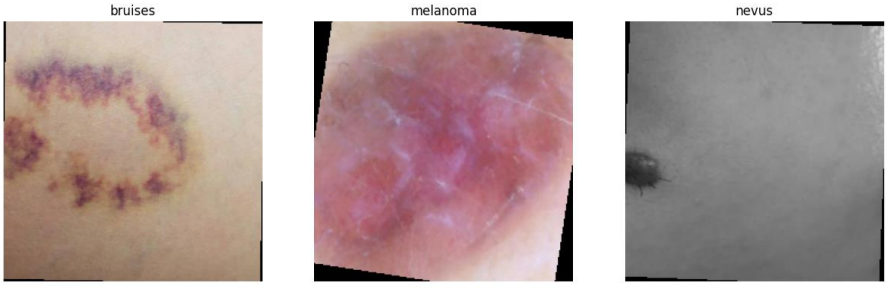
Normalizing the image data to match the expected input format for the pre-trained VGG16 model, which involved adjusting pixel values to a standard scale.

2. Data Loading:

We utilized the PyTorch ImageFolder class to load images and their corresponding labels from directories. Two separate data loaders were instantiated for training and testing purposes, with a batch size of 16 and shuffling enabled for the training set to ensure randomization.

3. Visualization:

Prior to model training, we visualized a subset of the data to confirm the effectiveness of our data augmentation strategies. This involved displaying a few sample images from each class and ensuring that they were correctly labeled.



4. Model Selection and Modification:

We chose the VGG16 architecture, a proven model in image classification tasks, as our starting point. Given that VGG16 is typically trained on a more extensive set of classes, we modified the final layer of the pre-trained model to output three classes corresponding to our dataset.

The feature extraction layers of the model were frozen to retain the learned patterns, and only the classifier layers were trained.

The model was transferred to a CUDA-enabled GPU for accelerated computation.

5. Training:

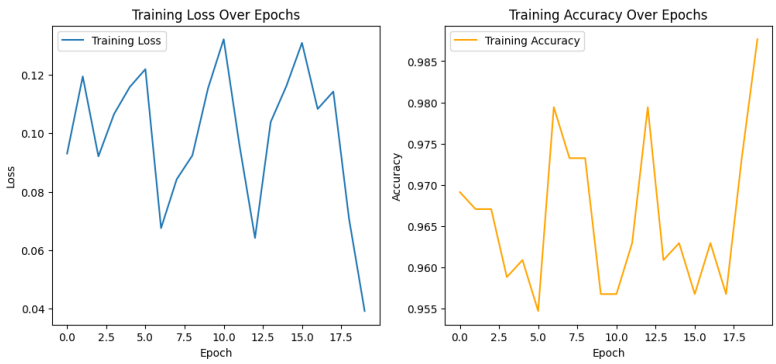
The model was trained over 20 epochs using the Adam optimizer with a learning rate of 0.0001. During each epoch, we monitored and recorded the training loss and accuracy to observe the learning progression and to adjust training parameters if needed.

6. Evaluation:

Post-training, the model's performance was evaluated on a separate test set. We calculated the average loss and accuracy to gauge the generalizability of the model. Additionally, a classification report was generated, providing detailed insights into precision, recall, and F1-scores for each class, which is crucial for medical diagnosis applications where the cost of false positives and negatives can be significant.

7. Results Visualization:

Upon completion of training and testing, we plotted the training loss and accuracy over the epochs to visually assess the model's learning behavior. Fluctuations in training accuracy suggested further tuning might be necessary to achieve more stable learning.



8. Performance Review:

The model exhibited high accuracy, with a test accuracy of 97.36%, and precision and recall metrics indicating reliable performance across all classes. However, the class 'nevus,' due to its smaller representation in the dataset, showed a lower recall, which indicates an area for potential improvement, possibly through techniques such as data augmentation or class re-weighting.

9. Conclusions:

The classification model demonstrated robust performance, suggesting its suitability for assisting in medical image analysis. Future work will explore cross-validation to validate the model's stability further and consider collecting more data for underrepresented classes to improve classification balance.

- **Dataset Sources:**
  - Nevus dataset:
    - https://universe.roboflow.com/aomsin-pvpaz/-vi0h8
  - Melanoma dataset:
    - https://universe.roboflow.com/dd-xovzc/cancer-sjstw
  - Bruises dataset:
    - https://universe.roboflow.com/new-workspace-khpun/bruises