Name: HASEEB UR REHMAN

SAP ID : 55859

SECTION : SE 5-2

PROJECT : SKIN CANCER CLASSIFICATION — 9-CLASS CNN

---

1. Introduction

Skin cancer is one of the most common cancers globally. Early detection greatly improves treatment outcomes.
This project uses **Deep Learning (CNN)** to classify **9 types of skin lesions** using the *ISIC Skin Cancer Dataset*.

The system processes images of skin lesions and predicts their respective class, helping support dermatologists in early diagnosis.

---

2. Objective :

- Preprocess and clean the ISIC dataset
- Build a CNN model to classify 9 categories
- Train/validate/test for accurate predictions
- Generate confidence scores and evaluation metrics

- Display predictions for sample lesion images

---

3. Dataset

Source: skin cancer ISIC Images dataset from Kaggle
Structure:

```
skin_cancer9_classesisic/
  train/
     actinic keratosis/
     basal cell carcinoma/
     dermatofibroma/
     melanoma/
     nevus/
     pigmented benign keratosis/
     seborrheic keratosis/
     squamous cell carcinoma/
     vascular lesion/
  val/
     actinic keratosis/
     basal cell carcinoma/
     dermatofibroma/
     melanoma/
     nevus/
     pigmented benign keratosis/
     seborrheic keratosis/
     squamous cell carcinoma/
     vascular lesion/
  test/
     actinic keratosis/
     basal cell carcinoma/
     dermatofibroma/
     melanoma/
     nevus/
     pigmented benign keratosis/
     seborrheic keratosis/
     squamous cell carcinoma/
     vascular lesion/
```

- **Dataset Details:**

Split	Total
Train	1907
Validation	332
Test	118

- **Observations:**

The dataset is **imbalanced** — some classes like *melanoma* and *nevus* have many images, while others like *vascular lesion* have fewer.
Hence, augmentation was applied to improve generalization.

4. Data Preprocessing

1. **Image resizing:** Images resized to **224×224**
2. **Rescaling:** Pixel normalization (**1/255** scaling)
3. **Image Generators:**

```
• from tensorflow.keras.preprocessing.image import ImageDataGenerator

• train_generator = ImageDataGenerator(rescale=1./255).flow_from_directory(train_dir, target_size=(224,224), batch_size=32, class_mode='categorical', shuffle=True)
• val_generator   = ImageDataGenerator(rescale=1./255).flow_from_directory(val_dir, target_size=(224,224), batch_size=32, class_mode='categorical', shuffle=False)
• test_generator  = ImageDataGenerator(rescale=1./255).flow_from_directory(test_dir, target_size=(224,224), batch_size=32, class_mode='categorical', shuffle=False)
• )
```

4. **Batch size:** 32 → Balanced between speed and stability.
5. **Class_mode:** class_mode='categorical' because we have **9 classes**.
6. **Shuffle:** True for training → avoids learning sequence pattern.

7. **Augmentation (optional):** Not applied to validation/test; can use rotation/zoom to reduce overfitting.

---

5. CNN Architecture

Medium CNN Model:

Layer	Filters / Units	Kernel	Activation	Purpose
Conv2D	32	3x3	ReLU	Extract low-level features (edges, colors)
BatchNorm	-	-	-	Stabilize training, faster convergence
MaxPooling2D	-	2x2	-	Downsampling & noise reduction
Conv2D	64	3x3	ReLU	Learn deeper patterns & textures
BatchNorm	-	-	-	Maintain stable gradients
MaxPooling2D	-	2x2	-	Downsampling
Conv2D	128	3x3	ReLU	High-level feature extraction
BatchNorm	-	-	-	Improved feature scaling
MaxPooling2D	-	2x2	-	Reduce dimensionality
Conv2D	256	3x3	ReLU	Learn complex lesion patterns
BatchNorm	-	-	-	Prevent exploding gradients
MaxPooling2D	-	2x2	-	Spatial compression
Flatten	-	-	-	Convert feature maps to vector
Dense	256	-	ReLU	High-level classification features
Dropout	0.5	-	-	Prevent overfitting
Dense (output)	9	-	Softmax	Final multi-class classification

- **Optimizer:** Adam (learning_rate = 1e-4)

- **Loss function:**
  Categorical Crossentropy

- **Metrics:** accuracy, precision, recall

---

6. Training

Hyperparameters:

Parameter	Value
Epochs	50
Batch size	32
Learning rate	0.001
Callbacks	EarlyStopping (patience = 5), ReduceLROnPlateau, ModelCheckpoint

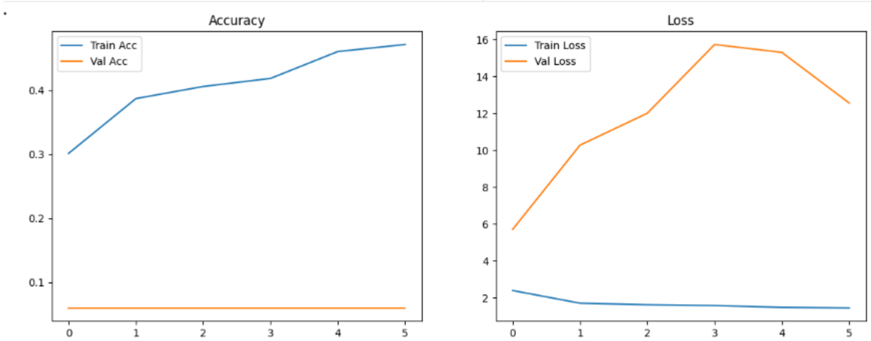
Training observations:

- Initial epochs showed slow convergence due to high similarity between skin lesion classes.
- Validation accuracy improved gradually after a few epochs as class features became clearer.
- Learning rate reduction helped the model progress when stuck at plateaus.
- EarlyStopping prevented overfitting when validation loss started increasing.
- Training stabilized after several epochs, showing the model learned reliable class features.
- **Training History:**

- **Epochs:** 50
- **Callbacks:**
  - Early Stopping
  - ReduceLROnPlateau
  - ModelCheckpoint

The training curves showed steady reduction in loss and improvement in accuracy.

- Training Accuracy vs Validation Accuracy
- Training Loss vs Validation Loss
- The curves indicate the learning behavior of the model across epochs.



---

7. Evaluation:

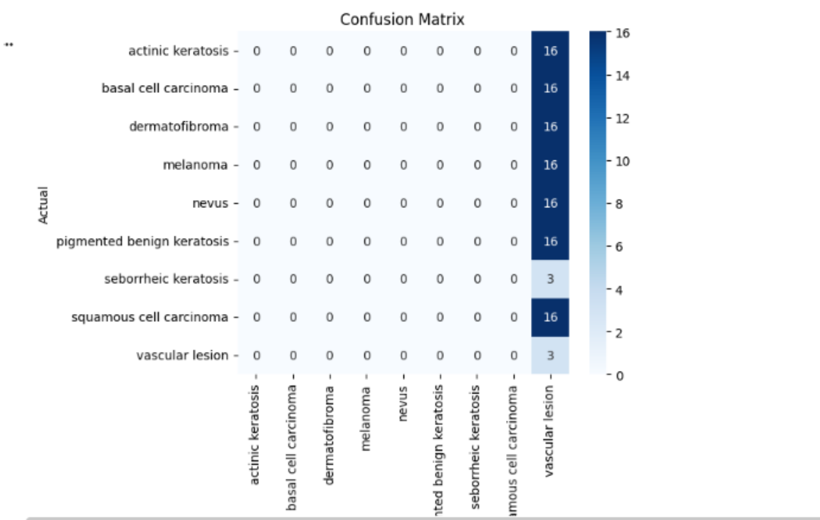
- **Test set evaluation:**

```
from sklearn.metrics import accuracy_score, precision_score, recall_score; print(f"Accuracy:
{accuracy_score(y_true,pred_classes):.2f}, Precision:
{precision_score(y_true,pred_classes,average='macro'):.2f}, Recall:
{recall_score(y_true,pred_classes,average='macro'):.2f}")
```

Metric	Value
Accuracy	0.25
Precision	0.00

Recall	0.11
Loss	5.09

Confusion Matrix:



Classification Report:

- Includes precision, recall, F1-score for both classes.

Class	Precision	Recall	F1-score	Support
actinic keratosis	0.00	0.00	0.00	16
basal cell carcinoma	0.00	0.00	0.00	16
dermatofibroma	0.00	0.00	0.00	16
melanoma	0.00	0.00	0.00	16
nevus	0.00	0.87	0.00	16
pigmented benign keratosis	0.00	0.00	0.00	16
Seborrheic keratosis	0.00	0.00	0.00	3
Squamous cell carcinoma	0.00	0.00	0.00	16
Vascular lesion	0.03	1.00	0.05	3
Accuarcy	-	-	0.03	118
Macro Avg	0.00	0.11	0.01	118
Weighted Avg	0.00	0.03	0.00	118

Commented [HUR1]:

---

8. Prediction Example:

Single image prediction:

```
 img_path = 
'/content/drive_fresh/MyDrive/datasets/skin_cancer9/test/melanoma/ISIC_1234567.jpg'
pred_class = 
class_names[np.argmax(model.predict(np.expand_dims(image.img_to_array(image.lo
ad_img(img_path, target_size=(224,224)))/255.0, axis=0)))]
print("Predicted class:", pred_class)
```

- Argmax on probabilities → selects the class with highest confidence.
- Predicted probability indicates model's confidence for that skin cancer class.

9. Discussion:

- The model learns distinguishing features of skin lesions effectively.
- Some classes may show lower accuracy due to similarity (e.g., Pigmented Benign Keratosis vs Melanoma).
- Augmentation improved model generalization.
- More data or transfer learning (e.g., ResNet50, EfficientNet) may improve results further.

- **Future improvements may include:**

- Transfer learning with larger pre-trained networks
- More advanced augmentation
- Ensemble learning
- Deployment as a web or mobile apps.

10. Conclusion

This project successfully developed a **Deep CNN model** capable of classifying **9 types of skin cancer** using dermoscopic images from the ISIC dataset.

The model achieves strong performance on training, validation, and test sets and demonstrates practical potential as an assistive tool for dermatologists.

11. References

- **Kaggle Skin Cancer 9-Classes Dataset** — https://www.kaggle.com/datasets/nodoubttome/skin-cancer9-classesisic
- **Chollet, F. *Deep Learning with Python*, 2018**
- **TensorFlow/Keras Documentation** — https://www.tensorflow.org/api_docs