A logo for a company

Description automatically generated

**Computer – Aided Diagnosis Final Project Report**

**Assigned for Dr. Rana Hossam ELDeen & TA.Heidi Ahmed**

**By the Team Members :**

**EzzEldien Magdy Fathy Hafez**

**Mostafa Mohamed Mostafa**

**Mohamed Mahmoud Shawky**

**Mazen Hany Abdelsalam**

**Tuqa Yasser Roshdy**

**Rawan Tarek Taha**

Report on Image Classification Using Machine & Deep learning Techniques

**Introduction**

For this report, the procedural method of classifying skin images using hierarchical wavelet transforms is highlighted. The dataset consists of lots of images with different skin features and the work is done using Python in Kaggle platform.

**1st approach**

**Step 1: Data Reading**

• Task: Retrieve images from the dataset of Monkeypox skin images categorized in four classes: chickenpox, measles, monkeypox, and normal.

• Details:

* Images were changed into greyscale to ease computation.
* Then save the data into a data frame with the classes and labels at a list.

**Step 2: GLCM Feature Extraction**

• Task: Use GLCM as a feature extraction method

• Details:

* Constructed Principals: ASM, Contrast, Correlation, Homogeneity, and Energy.
* Results: GLCM features were computed for each image in the dataset.

**Step 3: Feature Ranking Using Fisher’s Discriminant Ratio**

• Goal: Rank features in order of their class discrimination based on the Fisher Score.

• Details:

* Features and their respective labels were fused together.
* Fisher Scores of the features under the study were computed and arranged in descending order.

**Step 4: SVM Classifier Training with Polynomial Kernel**

* **Task:** Train a Support Vector Machine (SVM) classifier with a polynomial kernel.
* **Details:**
  + Split the dataset into training and testing sets.
  + Trained the SVM model with selected top features.
  + Evaluated the model using accuracy and a classification report.

**Step 5: Feature Extraction of Order N=1**

• Goal: Compute features based on histograms including Mean, Variance, Skewness, and Kurtosis.

• Details:

* Constructed a histogram from a flattened version of the image.
* Statistical measures were obtained using probabilities of intensity levels.

**Step 6: Feature Ranking Information Gain**

• Goal: Rank features using Information Gain in order of their importance.

• Details:

* Entropies and conditional entropies of each feature were determined.
* Features were ranked according to the Information Gain that was computed.

**Step 7: Applying SMOTE to Increase the Sample Size**

• Associated Activity: Enhance the representation of the minority class through oversampling utilizing SMOTE.

**Step 8: Using an SVM Classifier with a Polynomial kernel**

* Associated Activity: Build SVM model for classification using a polynomial kernel.

# 2nd approach

## **1. Data Preparation**

Steps:

* 1. Dataset Directory and Classes:  
  - Images are stored in subdirectories, each representing a specific class: Chickenpox, Measles, Monkeypox, and Normal.  
  - Classes are mapped to their respective image file paths.
* 2. Grayscale Conversion:  
  - Images are read and converted to grayscale using OpenCV (cv2).
* 3. Error Handling:  
  - Checks for missing or unreadable images.  
  - Skips images without detectable features.

Technique:

* - Basic file handling and image preprocessing using Python libraries (os, cv2).

## **2. Feature Extraction**

Steps:

* 1. Texture Features:  
  - Extracted using statistical texture analysis methods:  
   - GLCM (Gray Level Co-occurrence Matrix): Computes properties such as contrast, homogeneity, energy, correlation, etc.  
   - GLRLM (Gray Level Run Length Matrix): Extracts features based on runs of similar gray levels.  
   - GLSZM (Gray Level Size Zone Matrix): Measures zone-based texture properties.  
  - Libraries used: skimage.feature, mahotas.
* 2. Local Features:  
  - SIFT (Scale-Invariant Feature Transform):  
   - Detects keypoints in images.  
   - Extracts local descriptors invariant to scaling, rotation, and lighting changes.
* 3. Error Handling:  
  - Checks for images with no detectable keypoints or descriptors.

Technique:

* - GLCM, GLRLM, and GLSZM for texture-based global features.  
  - SIFT for local feature detection and descriptor extraction.  
  - Hybrid approach combining global and local features.

## **3. Feature Engineering**

Steps:

* 1. Handle Missing Values:  
  - Replace NaN values in feature matrices with 0.
* 2. Feature Standardization:  
  - Standardize features to have zero mean and unit variance using StandardScaler.
* 3. Feature Aggregation:  
  - SIFT descriptors are combined into a feature matrix.

Technique:

* - Preprocessing techniques to ensure feature matrices are suitable for machine learning models.

## **4. Feature Selection**

Steps:

* 1. Fisher's Score:  
  - Approximation using SelectKBest and mutual\_info\_classif to rank features based on their mutual information with the class label.
* 2. Recursive Feature Elimination (RFE):  
  - Selects the top n features iteratively by fitting a Logistic Regression model.
* 3. Selected Features:  
  - Outputs the top-ranked features for subsequent modeling.

Technique:

* - Filter-based and wrapper-based feature selection techniques.

## 5**. Data Balancing**

Steps:

* 1. Class Imbalance Handling:  
  - Applied SMOTE (Synthetic Minority Oversampling Technique) to oversample minority classes in the training dataset.
* 2. Split Data:  
  - Stratified train-test split ensures class proportions are maintained.

Technique:

* - SMOTE for synthetic data generation to balance class distributions.  
  - Stratified splitting to preserve class proportions.

## **6. Model Training and Evaluation**

Steps:

* 1. Train Models:  
  - Applied three different models:  
   - SVM (Support Vector Machine)  
   - Random Forest Classifier  
   - Logistic Regression  
  - Models are trained on balanced datasets.
* 2. Evaluate Models:  
  - Metrics computed:  
   - Accuracy: Measures overall performance.  
   - Classification Report: Provides precision, recall, F1-score, and support for each class.

Technique:

* - Supervised machine learning algorithms for classification tasks.  
  - Model evaluation using scikit-learn metrics.

## **7. Bag-of-Features with SIFT**

Steps:

* 1. Local Descriptor Aggregation:  
  - SIFT descriptors from all images are combined into a unified feature matrix.  
  - Descriptors are labeled according to their respective image classes.
* 2. DataFrame Construction:  
  - Creates a DataFrame with descriptors as features and their corresponding class labels.

Technique:

* - Bag-of-Features approach (preliminary step for methods like Bag-of-Words or Fisher Vectors).

A screenshot of a computer screen

Description automatically generated

## **8. Output and Results**

Outputs:

* **1. Top Features:**  
  - Features selected by Fisher's Score.

A black text on a white background

Description automatically generated

A screenshot of a computer

Description automatically generated- RFE

* **2. Classification Performance:**  
  - Accuracy and classification reports for each model.

A screenshot of a computer screen

Description automatically generated

**3. Feature Rankings:**  
- Ranked list of all features based on RFE.

A screenshot of a computer

Description automatically generated

# 3rd approach

**Step 1: Setting Up the Environment**

The environment is set up by importing necessary libraries and defining the working directory.

Libraries Used

The following libraries are utilized in this analysis:

NumPy: For numerical operations.

Pandas: For data manipulation and analysis.

OpenCV (cv2): For image processing.

Matplotlib: For data visualization.

Scikit-learn: For machine learning tasks.

imgaug: For image augmentation.

**Step 2: Loading the Dataset**

The dataset consists of classes and images classified accordingly are uploaded in a dictionary.

1. Measles
2. Monkeypox
3. Normal
4. Chickenpox

**Hierarchical Wavelet Transform:**

The hierarchical structure of wavelets is a powerful tool for feature extraction from images. This approach decomposes images into multiple sub-bands at varying levels of detail, enabling the analysis of spatial and frequency information. By leveraging this method, we can extract meaningful features that represent the texture and structure of images.

Methodology:

Each image is decomposed using a wavelet transform (e.g., Haar wavelets) into sub-bands: Low-Low (LL), Low-High (LH), High-Low (HL), and High-High (HH).

Decomposition is performed hierarchically across multiple levels, creating a nested structure of features.

Feature Extraction:

**4- Texture features are computed for each sub-band using the Gray-Level Co-Occurrence Matrix (GLCM).**

Key metrics include:

Contrast

Correlation

Energy

Homogeneity

These features are calculated for all channels of the image (e.g., RGB) independently.

**5- Data Augmentation:**

To balance the dataset, image augmentation techniques such as flipping, rotation, brightness adjustment, and blurring are applied.

Augmented images are kept in memory to avoid unnecessary disk I/O.

**6- Feature Consolidation:**

Extracted features from all levels and channels are flattened and combined into a structured DataFrame.

The DataFrame includes features and their corresponding labels, enabling seamless analysis and model training.

**7- Classification and Analysis:**

Features are fed into a machine learning model (e.g., Random Forest) for classification.

The performance is evaluated using metrics like the confusion matrix and classification report.

Expected Output

**8- The output is a DataFrame containing:**

Extracted features for each image at all hierarchical levels.

Labels corresponding to the images for supervised learning.

A balanced representation of the dataset achieved through augmentation.

**Key Results**

A balanced dataset was created, mitigating class imbalance issues.

High-dimensional features representing both spatial and frequency information were extracted for robust analysis.

**The pipeline demonstrated effectiveness in generating a DataFrame suitable for machine learning and analytical tasks.**

**Step 6: Visualization**

It is possible to visualize images that belong to certain categories by randomly selecting them from each class.

Expected Output

Visual inspection of the dataset is possible by displaying a grid of images of different classes.

# 4th approach

**Deep learning partition**

**Data Preparation**

1. **Data Collection**: Images for each class were collected and stored in respective directories.
2. **Data Preprocessing**:
   * All images were loaded into a list and their corresponding labels into another.
   * The images were converted into NumPy arrays for computational efficiency.
   * Labels were one-hot encoded to prepare them for classification tasks.
3. **Data Splitting**: The dataset was split into training and testing subsets.
4. **Data Augmentation**:
   * A data generator was created for the training set with transformations such as rotation, width and height shifting, shearing, zooming, and horizontal flipping.
   * The validation set was prepared without any augmentation to maintain evaluation consistency.

**Using VGG16 as a Feature Extractor**

1. **Feature Extraction**:
   * The VGG16 model, pre-trained on ImageNet, was used to extract feature maps from the images.
   * The top (classification) layers of VGG16 were removed, and only the convolutional base was utilized.
2. **Classification with Random Forest**:
   * The extracted features were passed to a Random Forest classifier.
   * The model achieved an accuracy of 89.25%.
   * Precision, recall, and F1-score metrics indicated good performance, particularly for the "Monkeypox" and "Normal" classes.
   * The confusion matrix highlighted some misclassifications between "Chickenpox" and "Measles."
3. **Classification with SVM**:
   * Similarly, the extracted features were input to an SVM classifier.
   * The model achieved an accuracy of 85.60%.
   * Precision and recall metrics were slightly lower compared to the Random Forest classifier, especially for the "Measles" class.

**Using VGG16 with Transfer Learning**

1. **Model Architecture**:
   * The VGG16 model was loaded with pre-trained ImageNet weights, excluding the top layers.
   * A dense layer was added to adapt the model to the four-category classification task.
2. **Training**:
   * The modified model was trained on the prepared dataset.
   * Early stopping and learning rate adjustments were applied to optimize training.
3. **Evaluation**:
   * The model achieved a training accuracy of 94.80% and a validation accuracy of 92.50%.
   * Precision, recall, and F1-scores were consistently high across all classes.
   * Training and validation accuracy/loss curves showed a smooth convergence without significant overfitting.

**Results and Discussion**

1. **Feature Extraction**:
   * Random Forest Classifier:
     + Accuracy: 89.25%
     + Highlighted strong performance for "Monkeypox" and "Normal" categories.
   * SVM Classifier:
     + Accuracy: 85.60%
     + Slightly lower recall for "Measles."
2. **Transfer Learning**:
   * VGG16-based model:
     + Training Accuracy: 94.80%
     + Validation Accuracy: 92.50%
     + Precision and recall metrics indicated consistent performance across all classes.
3. **Comparison**:
   * Transfer learning provided the best overall performance, especially in terms of accuracy and recall.
   * Feature extraction with Random Forest offered a competitive alternative for simpler classification tasks.