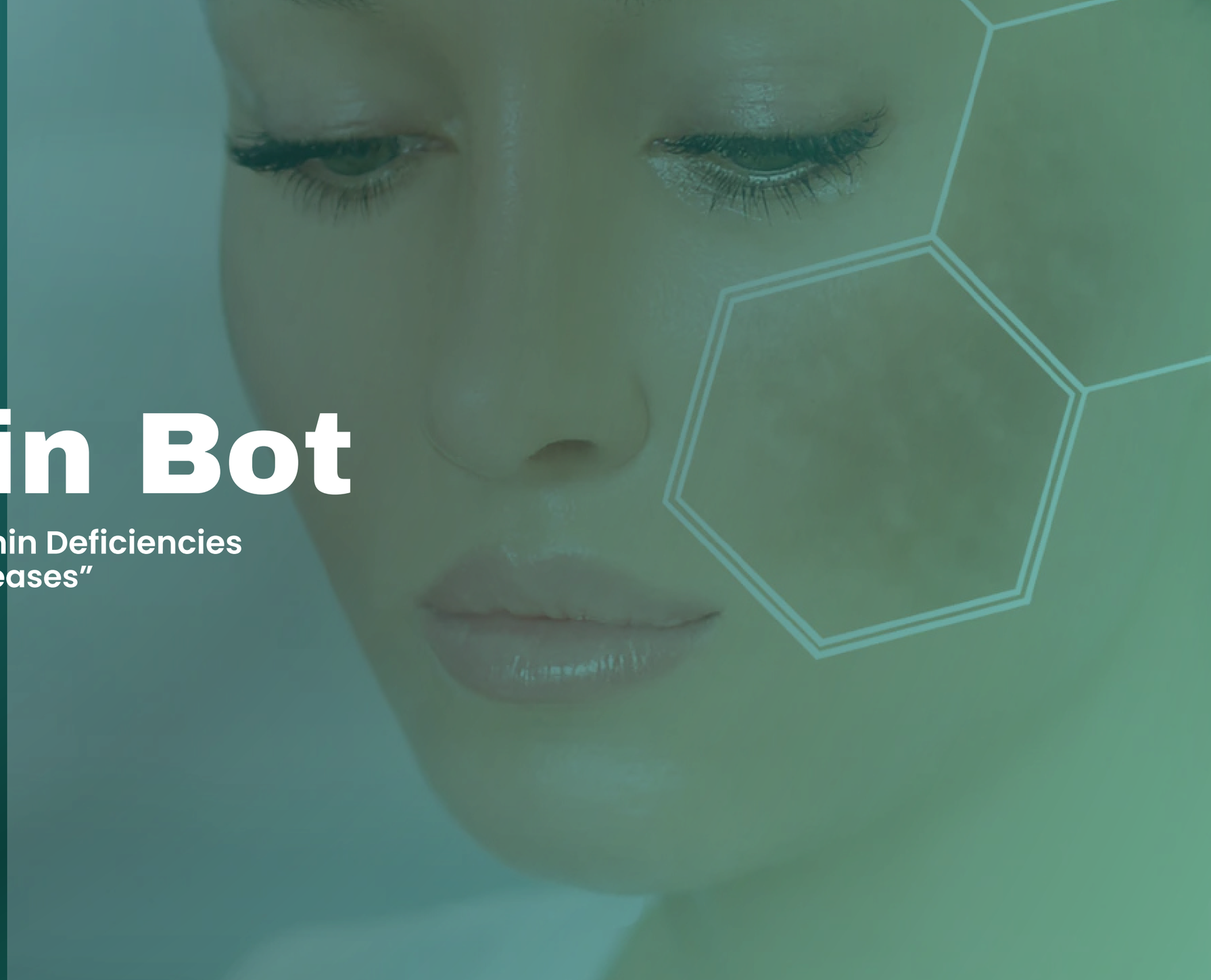
NutriSkin Bot

"The Impact of Vitamin Deficiencies on Skin Diseases"

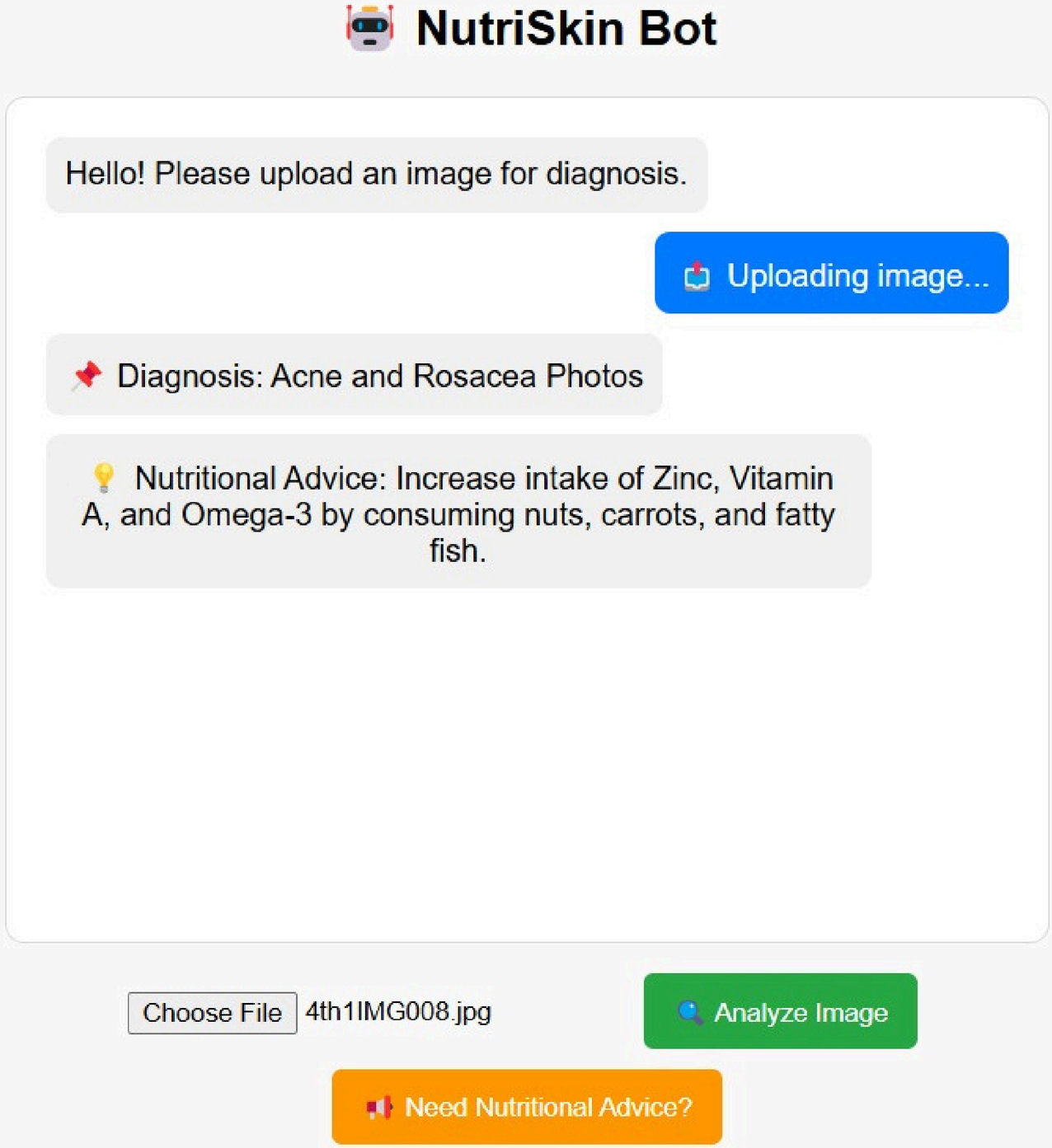


Exploring project Objective

Skin health is deeply influenced by nutrition. Vitamins play a crucial role in maintaining skin integrity, hydration, and protection against diseases. Deficiencies in essential vitamins can lead to various skin conditions such as acne, eczema, pigmentation disorders, and psoriasis.

This project explores how AI-based classification models can detect and analyze skin conditions linked to vitamin deficiencies, aiding in early diagnosis and prevention.
pabilities.

Flak Web App


NutriSkin Bot
Hello! Please upload an image for diagnosis.
Uploading image...
Diagnosis: Acne and Rosacea Photos
Nutritional Advice: Increase intake of Zinc, Vitamin A, and Omega-3 by consuming nuts, carrots, and fatty fish.
Choose File
4th1IMG008.jpg
Analyze Image
Need Nutritional Advice?

Methodolegy:

Data Used:

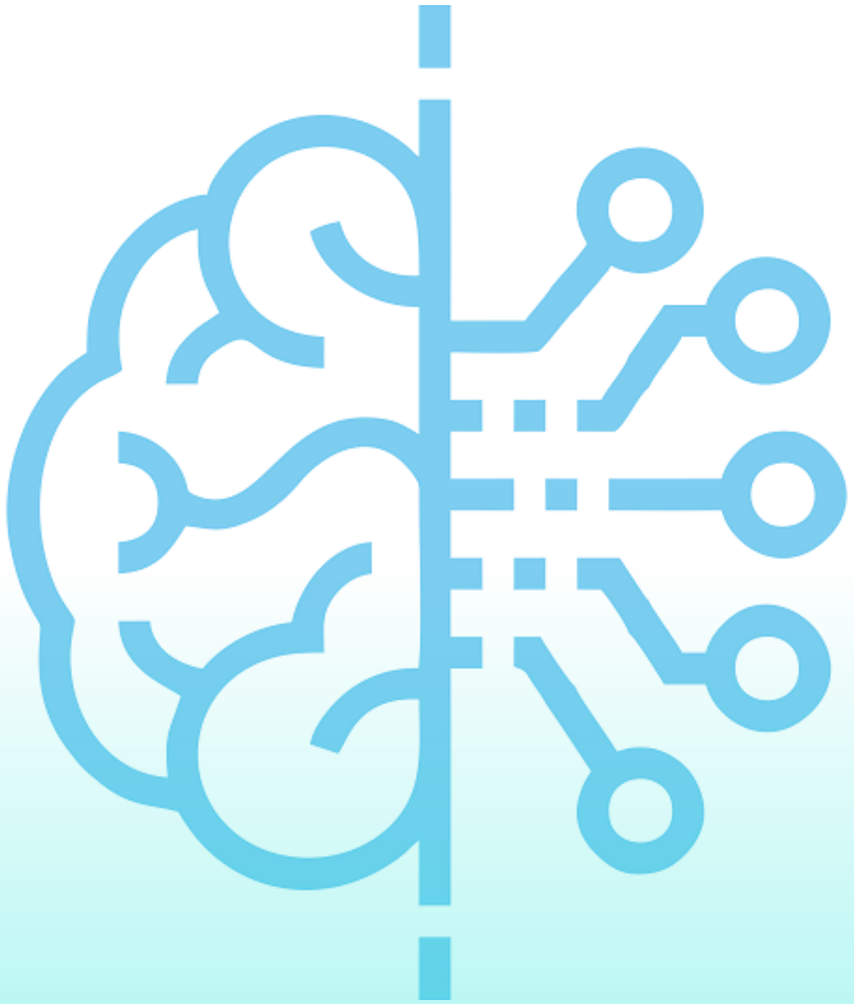
Skin disease images and their relation to vitamin deficiencies. (Dermnet dataset)

Model Used:

CNN based Models

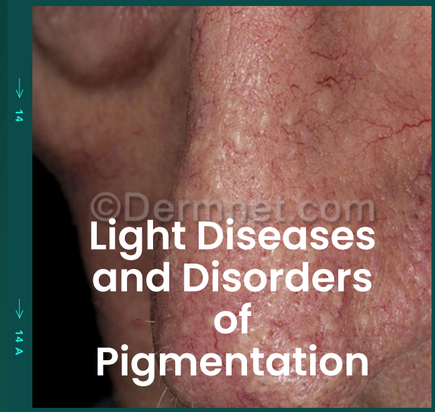
Classification Approach

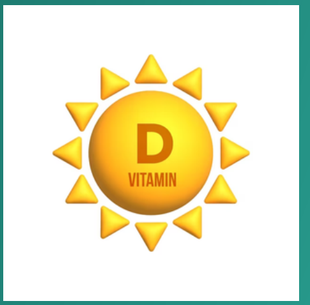
How does the model predict skin conditions?



Targeted Skin Conditions

Top 5 most affected skin diseases :.





Model Building Steps

Model Selection ?

ResNet152 Vs mobileNetV3

ResNet152

-WHY?

-Steps?

-Results?

Image processing :

```
# Image Data Generators
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=40,
    width_shift_range=0.3,
    height_shift_range=0.3,
    shear_range=0.3,
    zoom_range=0.3,
    brightness_range=[0.8, 1.2],
    horizontal_flip=True,
    fill_mode='nearest'
)

test_datagen = ImageDataGenerator(rescale=1./255)

# Load Images

def create_data_generator(df, datagen, directory):
    return datagen.flow_from_dataframe(
        df, x_col='path', y_col='class', target_size=(224, 224),
        batch_size=32, class_mode='categorical'
    )

train_generator = create_data_generator(dermnet_train_df, train_datagen,
test_generator = create_data_generator(dermnet_test_df, test_datagen, test_dir)
```

```python
from tensorflow.keras import layers

# Build Model
base_model = ResNet152(weights='imagenet', include_top=False, input_shape=(224, 224,
base_model.trainable = True  # Enable fine-tuning

# Freeze initial layers and train only last 50 layers
for layer in base_model.layers[:-50]:
    layer.trainable = False

model = keras.Sequential([
    base_model,
    layers.GlobalAveragePooling2D(),
    layers.Dense(256, activation='relu'),
    layers.Dropout(0.5),
    layers.Dense(n_classes, activation='softmax')
])
```

Model Structure

```
# Callbacks
callbacks = [
    EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True),
    ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=3, min_lr=1e-
6)
]
# Train Model
model.fit(
    train_generator,
    validation_data=test_generator,
    epochs=10,
    class_weight=class_weights_dict,
    callbacks=callbacks
)
```

Model Training

Bad Accuracy

High Computations

MobileNet V3

-WHY?

-Steps?

-Results?

Image processing

✅preprocess_input ensures the input images match the format used during the model's training, improving accuracy and consistency.

✅ For MobileNetV3, it scales pixel values from [0,255] to [-1,1], optimizing the input for better model performance.

```
# Data preparation
batch_size = 32
img_size = (224, 224)
data_dir = train_dir

datagen = ImageDataGenerator(
    validation_split=0.2,
    preprocessing_function=tf.keras.applications.mobilenet_v3.preprocess_input,
)

# Training data generator
train_generator = datagen.flow_from_directory(
    data_dir,
    target_size=img_size,
    batch_size=batch_size,
    class_mode='categorical',
    subset='training',
    classes=classes
)
```

```
base_model = MobileNetV3Large(weights=None, include_top=False, input_shape=(224, 224, 3))
base_model.load_weights(local_weights_path)
#Freeze the early layers of the base model
for layer in base_model.layers[:10]:
    layer.trainable = False

# Build a new model on top of the pre-trained base
inputs = base_model.input
x = base_model(inputs, training=False)
x = GlobalAveragePooling2D()(x)
x = Dense(512, activation='relu')(x)
x = Dropout(0.5)(x)
outputs = Dense(num_classes, activation='softmax')(x)

model = Model(inputs, outputs)
```

Model Structure

```
#Callback to save the best model. Using checkpoint and earlystopping to monitor validation accuracy
callbacks_list = [
    tf.keras.callbacks.ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=3, min_lr=1e-6),

    tf.keras.callbacks.ModelCheckpoint(
        filepath='mbNetV2model2.keras',
        monitor='val_accuracy', save_best_only=True, verbose=1),
    tf.keras.callbacks.EarlyStopping(monitor='val_accuracy', patience=10,verbose=1)
]
history = model.fit(train_generator,
                    epochs=50,
                    validation_data=val_generator,
                    # class_weight=class_weights_dict,
                    callbacks=[callbacks_list])
```

Model Training

- Lightweight
- fast
- High accuracy

```
test_loss, test_accuracy = model.evaluate(test_gen)
print(f'Test Accuracy: {test_accuracy * 100:.2f}%')
```

```
142/142 ━━━━━━━━━━━━━━━━━━━━ 21s 144ms/step - accuracy: 0.9194 - loss: 0.4406
Test Accuracy: 92.31%
```

Problems During Learning

- Features Selection.
- Bad image preprocessing leads to bad accuracy.
- Testing data also needs preprocessing.
- High computations for ResNet model.

Future plan

- Expand the Number of Predicted Diseases : Currently, the model classifies 10 skin conditions, but we aim to increase the number to cover a wider range of vitamin-related skin diseases.
- Add a ‘Normal Skin’ Category → To differentiate between diseased and healthy skin, improving model accuracy and reducing false positives.
- Collaborate with dermatologists to verify AI predictions and refine the model.

Check Our Work :

Kaggle

Github

References :

- Vitamins Deficiency Detection Using Image Processing and Neural Network
- Vitamin Deficiency Detection Using ImageProcessing and Neural Network
- Kaggle Notebook
- Kaggle notebook
- Dermnet dataset

National Telecommunication Institute

HUAWEI

Supervisor :

Dr Elhossiny

Zainab Ayman

Rahma Osama

Thank You