

Malaria Infected Cell Classification

Your Name

PULICHERU PRASHANTH

INTRODUCTION

This technical report presents an analysis of the Malaria-Infected Cell Classification project. The primary objective of the project is to develop a deep learning model capable of classifying cells as either infected with malaria or uninfected. The model is trained on a dataset consisting of cell images obtained from the Kaggle Malaria Cell Images dataset.

Approach

We utilized a Convolutional Neural Network (CNN) for the classification task. The CNN architecture consists of two convolutional layers, each followed by max-pooling layers to capture hierarchical features from the input images. The network is then flattened, and two fully connected layers are added for further feature extraction and classification. The final layer uses the sigmoid activation function to output a probability

indicating the likelihood of a cell being infected.

```
# Define the model
model = models.Sequential([
    layers.Conv2D(32, (3, 3), activation='relu', input_shape=(128, 128, 3)),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dense(1, activation='sigmoid')
])

# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

Data Preprocessing

We employed the `ImageDataGenerator` from TensorFlow to preprocess the training data. Image augmentation techniques such as shearing, zooming, and horizontal flipping were applied to enhance the model's ability to generalize to unseen data.

```
# Create data generators
train_datagen = ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True)

train_generator = train_datagen.flow_from_directory(
    'D:\\Downloads\\Dataset\\cell_images\\Train',
    target_size=(128, 128),
    batch_size=32,
    class_mode='binary',
    classes=['uninfected', 'parasite'], # Specify the class names
    shuffle=True # Set to True for better training
)
```

Training

The model was trained for 5 epochs using the generated training data.

```
# Train the model
model.fit(train_generator, epochs=5)
model.save('malaria_model.h5')
```

Findings

Model Evaluation

The trained model achieved satisfactory accuracy on the training dataset. Further evaluation on a separate test dataset is recommended to assess the model's generalization performance. The accuracy metric during training can be visualized over epochs to observe convergence.

Streamlit Application

A Streamlit application was developed to allow users to upload cell images and receive predictions from the trained model. The application displays the uploaded image, its resized version, and the model's predictions for both infected and uninfected classes.

```
# Streamlit app
st.title('Malaria Infected Cell Classification')
uploaded_file = st.file_uploader('Upload an image', type=['jpg', 'jpeg', 'png'])

if uploaded_file is not None:
    st.image(uploaded_file, caption='Uploaded Image.', use_column_width=True)
    st.write('')
    st.write('Classifying...')

    # Call the function to predict and overlay hotspots
    predict_and_overlay(uploaded_file)
```

Recommendations

Model Evaluation: Evaluate the trained model on a separate test dataset to obtain a more accurate assessment of its performance and generalization ability.

Fine-tuning: Consider fine-tuning the model architecture or hyperparameters to improve overall performance.

Deployment: Explore options for deploying the model in a production environment, considering factors such as latency, scalability, and security.

User Interface: Enhance the Streamlit application with a user-friendly interface and additional features for better usability.

Conclusion

This technical report outlines the approach, findings, and recommendations for the Malaria-Infected Cell Classification project. The development of a deep learning model for cell classification demonstrates promising results, with opportunities for further improvement and deployment in real-world scenarios.

REFERENCES

1. <https://lhncbc.nlm.nih.gov/LHC-downloads/downloads.html#malaria-datasets>
2. <https://www.kaggle.com/iarunava/cell-images-for-detecting-malaria>