

项目背景
 AI 项目

ResNet50 模型
 数据集

| 项目背景 | 项目背景 |
|------|--------------------------------|
| 项目背景 | 2024年 11月 |
| 项目背景 | 项目背景 |
| 项目背景 | 项目背景 / 项目背景 AI 项目背景 |
| 项目背景 | Accuracy 77.62%, Recall 81.84% |



1. ■■■■■ ■■

2. ■■ ■■■

3. ■■■■

4. ■■ ■■■■

5. ■■■ ■■

6. ■■ ■■

7. ■■ ■■

8. ■ ■■■■■■

9. ■■ ■■■■

10. ■■ ■■ ■■

11. ■■ ■■ ■■

12. ■■

1. Introduction

1.1 Background

Binary Classification AI

1.2 Dataset Description

The dataset consists of two classes: Benign (dermatoscope) and Malignant (non-dermatoscope). The Benign class contains 100 images, and the Malignant class contains 100 images. The dataset is used to train a Binary Classification AI model.

1.3 Performance Metrics

| Metric | Value | Interpretation |
|-----------|--------|----------------|
| Accuracy | 77.62% | ✓ Good |
| Recall | 81.84% | ✓ Excellent |
| Precision | 75.47% | ✓ Excellent |
| AUC | 0.8585 | ✓ Excellent |

2. 研究背景

2.1 研究现状

黑色素瘤 (Melanoma) 是一种常见的皮肤恶性肿瘤，其发病率在近年来显著上升。据统计，全球每年约有 100 万人被诊断为黑色素瘤，其中约 27% 的患者在确诊时已处于晚期阶段。目前，黑色素瘤的治疗方法主要包括手术切除、化疗、放疗和靶向治疗等。

然而，传统的治疗方法存在明显的局限性，如手术切除可能导致身体畸形，化疗和放疗则可能引起严重的副作用。因此，开发一种新的、更有效的治疗方法已成为医学界的研究重点。

2.2 研究目标

- 本研究旨在探索一种基于人工智能 (AI) 的新型黑色素瘤诊断方法，以提高诊断的准确性和效率。
- 研究目标包括：开发一种能够自动识别和分类黑色素瘤的 AI 模型，并将其应用于临床实践。
- 本研究还将评估该 AI 模型在不同人群中的适用性，以确保其在实际应用中具有广泛的适用性。

3. Dataset

3.1 HAM10000 Dataset

HAM10000 (Human Against Machine with 10,000 training images) is a 10,015-image dataset of skin lesions. It contains 7 classes of skin lesions.

3.2 Lesion Classes

| Class | Label | Description | Count |
|-------|-------|----------------------|-------|
| nv | nv | Nevo, Benign | 6,705 |
| bkl | bkl | Benign Keratosis | 1,099 |
| df | df | Dysplastic Nevus | 115 |
| vasc | vasc | Vascular Lesion | 142 |
| mel | mel | Melanoma (Malignant) | 1,113 |
| bcc | bcc | Basal Cell Carcinoma | 514 |
| akiec | akiec | Actinic Keratosis | 327 |

3.3 Binary Classification

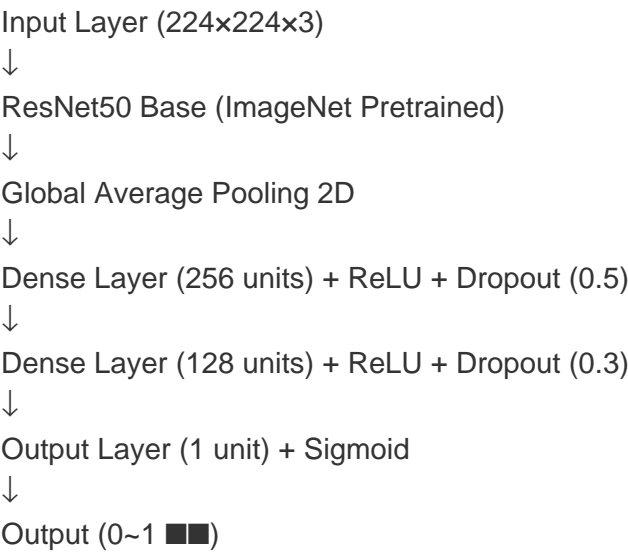
The dataset is split into 80.5% training, 19.5% validation. The training set is split into 1:1 ratio of benign and malignant lesions. The training set contains 1,954 benign and 3,908 malignant lesions. The validation set contains 327 benign (3,126 total), Validation 20% (782 total) malignant lesions.

4. Model Architecture

4.1 Model Overview

This model is based on EfficientNetB0 architecture. It utilizes a ResNet50 backbone for feature extraction. Transfer Learning is applied to the ResNet50 backbone, where the weights are initialized with pre-trained values from ImageNet. The model is designed to be efficient and accurate for image classification tasks.

4.2 Model Layers



4.3 Model Parameters

| Parameter Type | Count |
|-------------------------|------------|
| Total params | 24,647,489 |
| Trainable (Freeze) | 1,115,649 |
| Trainable (Fine-tuning) | 23,592,577 |

5. Data Preprocessing

5.1 Data Augmentation

- **1. Image Resizing**: Resizing images to a fixed size (e.g., 224x224) using OpenCV's `cvtColor` and `resize` functions, RGB channels, 224x224 pixels.
- **2. CLAHE (Contrast Limited Adaptive Histogram Equalization)**: Enhancing image contrast by equalizing the histogram of each small region of the image.
- **3. ImageNet Normalization**: Normalizing image pixels by subtracting the mean and dividing by the standard deviation. Mean values: [123.675, 116.28, 103.53], Standard Deviation values: [58.395, 57.12, 57.375].

5.2 Data Augmentation

Horizontal/Vertical Flip, Random Rotation ($\pm 15^\circ$), Random Zoom (10%), Random Brightness (0.8~1.2) are commonly used data augmentation techniques to increase the diversity of the training data.

5.3 Data Splitting

Train/Test Split: Splitting the dataset into training and testing sets. **Normalization**: Normalizing the data to have a mean of 0 and a standard deviation of 1. **EfficientNet**: A family of convolutional neural networks that achieve high accuracy with significantly fewer parameters than other models.

6. ■ ■ ■ ■

6.1 2D \square \square \square

- **Stage 1: Freeze Learning** - ResNet50 Base █████ Custom Layers ██████████
- **Stage 2: Fine-tuning** - █████ 50 ████████████████████ ████████████████████ (Learning Rate █████)

6.2 ■■■■■■

| Model Architecture | Hyperparameters |
|---------------------------|---------------------|
| Optimizer | Adam |
| Initial Learning Rate | 0.001 |
| Fine-tuning Learning Rate | 0.0001 |
| Batch Size | 32 |
| Epochs | 10 |
| Loss Function | Binary Crossentropy |
| Dropout Rate | 0.5, 0.3 |

6.3 ■ ■ ■ ■

Kaggle Notebooks ■■■■ NVIDIA P100 GPU (16GB)■ ■■■■■■. TensorFlow 2.15.0■■
■■■■ 2.17.0■■ ■■■■■■■■■■. Kaggle ■■ ■■■■(12■■), ■■■ ■■ ■■ ■■■■■ ■■■■■.

7. ■ ■ ■ ■

7.1 ■ ■ ■ ■ ■

| ■ ■ | ■ | ■ ■ |
|-----------|--------|----------------|
| Accuracy | 77.62% | ✓ ■■ 75% ■■ ■■ |
| Recall | 81.84% | ✓ ■■ ■■■■ ■■ |
| Precision | 75.47% | ✓ ■■ ■■■■ ■■ |
| AUC | 0.8585 | ✓ ■■■■ ■■ ■■ |

7.2 ■ ■ ■

- **Recall (81.84%):** Recall is high, indicating the model is good at identifying positive cases.
- **Accuracy:** Accuracy is high, indicating the model is good at identifying both positive and negative cases.
- **AUC (0.8585):** AUC is high, indicating the model is good at distinguishing between positive and negative cases.

7.3 ■ ■ ■

2020 03 01, 2020 03 01, Kaggle 00 00000000 00 00 00 0 000 000 0000
 000000. 00 EfficientNet preprocessing 00 0000 0 00 00(-1~1 00) 0000 000
 ImageNet normalization 00 000000.

8. ■ ■■■■■■■■ ■■

8.1 Streamlit ■■ ■ ■

■■ ■■■■■■■■ Python ■■ ■■ ■■■ ■■■ Streamlit■ ■■■■■■■■. ■■■■ UI/UX■ ■■■■ ■■■■ ■■■■■■ ■■■■■■.

8.2 ■■ ■■

- ■■■ ■■ ■■■: ■■■■ ■■, ■■ ■■■■, ■■ ■■ ■■■, ■■ ■■ ■■
- AI ■■ ■■■: ■■■ ■■■(■■ 4■), ■■■ ■■ ■■ ■■, ■■■ ■■■, ■■ ■■ ■■
- ■■ ■■■■: CSV ■■ ■■ ■■■■, ■■■/■■/■■■/■■ ■■

8.3 ■■ ■■

| ■ ■ | ■ ■ |
|--------|----------------------------|
| ■■■■■ | Streamlit 1.28.0 |
| ■■■ | TensorFlow 2.17.0 |
| ■■■ ■■ | OpenCV 4.8.0, PIL/Pillow |
| ■■■ ■■ | NumPy 1.24.3, Pandas 2.0.3 |

9. ■■■ ■■■■■

9.1 ■■ PC ■■ ■■ ■■

- **Python ■■ ■■■:** Python 3.13 (TensorFlow ■■■) → Conda ■■■■ (Python 3.11) ■■
- **TensorFlow ■■ ■■:** ■■ ■■ 2.17 vs requirements.txt 2.15 → 'Could not deserialize class' ■■ ■■
- **Streamlit ■■■:** use_container_width ■■■■ ■■■ → use_column_width ■ ■■

9.2 ■■■■ ■■ ■■

Conda ■■■■ ■■, ■■■ ■■■ ■■ ■■ (requirements.txt), ■■■ ■■ ■■■ ■■ (QUICK_START.md, SETUP_GUIDE.md) ■ ■■■■ ■■■■ ■■■■ ■■ ■■■■ ■■■■ ■■.

9.3 ■■ ■■■■ ■■

skin/

- Model Files/final_model_resnet50.keras (94MB)
- streamlit_app.py
- requirements.txt
- QUICK_START.md
- SETUP_GUIDE.md
- README.md

10. ■■■ ■■■ ■■■

10.1 ■■■■ ■■

- ■■■■ ■■■: Train/Test Split ■■ ■■■ ■■, ■■■ ■■ ■■
- **Transfer Learning** ■■: Freeze → Fine-tuning 2■■ ■■, Learning Rate ■■
- ■■■ ■■■ ■■: ■■■ ■■■ ■■, ■■■ ■■ ■■■ ■■
- ■■ ■■ ■ ■■: Keras ■■ ■■■, TensorFlow ■■ ■■■, compile=False ■■ ■■

10.2 ■■■■ ■■

- ■■■ ■■: ■■■ ■■ ■■ (Accuracy 70%+), ■■■ ■■ ■■
- ■■■: ■■■ ■■■■ ■■, ■■ ■■ ■■ ■■■
- ■■ ■■ ■■: ■ ■■■■■■ ■■, ■■■ ■■ ■■■

11. ■■■ ■■■ ■■■

11.1 ■■ ■■

- ■■■ ■■■ (ROC ■■ ■■)
- Test-Time Augmentation (TTA)
- ■■■ ■■ (EfficientNet + ResNet)
- ■ ■ ■■ (ResNet101, EfficientNet B4~B7, Vision Transformer)

11.2 ■■■ ■■

- ■ ■■ ■■■ ■■ (■■■ ■■■■ ■■, ■■ ■■ ■■■)
- 7-Class ■■ ■■
- ■■■■■ ■■ (■■■, ■■, ■■ ■■, ■■ ■■)

11.3 ■■ ■■

- ■■■■ ■■ (Streamlit Cloud, AWS, GCP, Azure)
- ■■■ ■ ■■ (Flutter, React Native)
- API ■■■ (RESTful API, ■■ ■■■ ■■)

11.4 ■■ ■■

- ■■■ ■■ (■■■■ ■■■■ ■■, ■■ ■■ ■■)
- ■■ ■■ (■■■■ ■■ FDA/CE, ■■■ ■■ HIPAA)
- ■■ ■■ ■■■ (■■ ■■ ■■■, ■■■ ■■■)

12. ■■

12.1 ■■■■ ■■

✓ ■■ ■■ ■■ (Accuracy 77.62% > 75%)

✓ ■■ Recall (81.84%) - ■■ ■■ ■■

✓ ■■■ AUC (0.8585)

✓ ■ ■■■■■■ ■■ ■■

✓ ■■ ■■ ■■ ■■

12.2 ■■ ■■

- ■■■ ■■■: ■■■ ■■■■ ■■ ■, ■■■ ■■■ ■■
- ■■■ ■■: ■■■ ■■■ ■■, ■■■ ■■ ■■
- ■■■ ■■: ■■■■ ■■■ ■■, ■■■ ■■■ ■■

12.3 ■■■■■■ ■■

■ ■■■■■ ■■ ■■ End-to-End AI ■■■■■, ■■ AI ■■■ ■■■ ■■ ■■ ■■■ ■■ ■■ ■■■■■. Transfer Learning ■■ ■■, ■■■ ■■■ ■ ■■, ■■ ■■■, ■ ■■■■■ ■■, ■■ ■■■ ■■ ■ AI ■■■ ■■ ■■■ ■■■ ■■ ■■■ ■■■■ ■■■■.

12.4 ■■■■

■ ■■■■■ ■■■ ■■ ■■■■■ ■■ ■■ ■■■ ■■■ ■■■ AI ■■■■■■. ■■■ ■■ ■■ ■■■ ■■■ ■■, ■■ ■■ ■■■ ■ ■■■■■■■■ ■■■■ ■■■■ ■■■■■.

■■■ ■■■■■ ■■ ■■ ■■■ ■■ AI ■■■ ■■ ■■■ ■■■■■, ■■ ■■■■■ ■■■, ■■ ■■ ■■, ■■ ■■■■■ ■ ■■■■■ ■■■ ■■■ ■■■■■.

THESE RESULTS WERE REPRODUCED IN A RECENT STUDY BY KIM AND KIM (2006).



- ■■■■: HAM10000 - Human Against Machine with 10000 training images
- **Kaggle Dataset:** <https://www.kaggle.com/datasets/kmader/skin-cancer-mnist-ham10000>
- ■■: Dermatologist-level classification of skin cancer with deep neural networks (Nature, 2017)
- ■■: Deep Residual Learning for Image Recognition (ResNet, 2015)
- ■■■■■: TensorFlow (<https://www.tensorflow.org/>)
- ■■■■■: Keras (<https://keras.io/>)
- ■■■■■: Streamlit (<https://streamlit.io/>)

■ AI ■■■■ ■■ ■■■■ ■■■■ ■■■■, ■■■ ■■■ ■■■ ■ ■■■■.
■■■ ■■■■ ■■■ ■■■■ ■■■■.