

**Group 17**  
**LEO1**

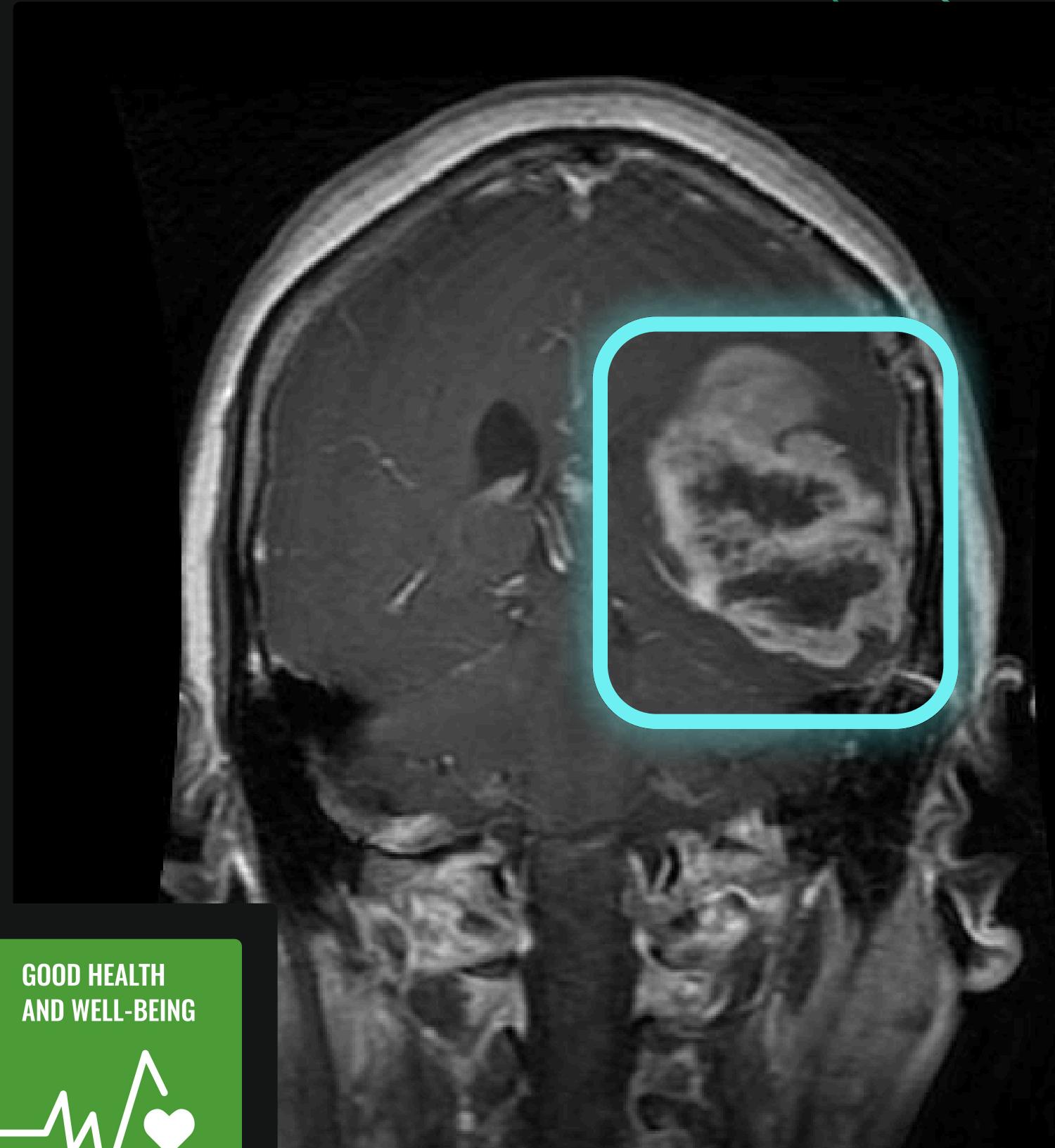
# BrainSight

**EFFicient Brain Tumor  
Segmentation Utilizing  
Computer Vision**

Kent Amadeo Timotheus - 2702227025

Theodore Zachary - 2702244100

Albertus Edbert Chandrajaya - 2702345440



# LATAR BELAKANG

Tumor otak merupakan penyakit akibat pertumbuhan sel abnormal pada otak, dengan lebih dari **150 ribu** kasus setiap tahun di **Indonesia**. Diagnosis melalui MRI masih bergantung pada analisis manual spesialis, sehingga prosesnya lambat dan kurang efisien. Dengan kemajuan **Computer Vision**, citra MRI dapat dianalisis secara otomatis untuk mengenali pola visual yang terkait tumor otak. Teknologi ini memungkinkan proses segmentasi menjadi lebih cepat, konsisten, dan mendukung dokter dalam pengambilan keputusan.

# Tentang Model Kami

- Model kami dirancang untuk mensegmentasi tumor otak berdasarkan citra MRI, dengan menggunakan AI. Model ini mampu mensegmentasikan berbagai tipe tumor otak, seperti **glioma, meningioma, atau pituitary**.
- Dengan menggunakan **machine learning**, model kami berfungsi sebagai alat bantu diagnosis yang dapat membantu tenaga medis untuk mengidentifikasi lokasi tumor. Pendekatan ini memungkinkan proses pemeriksaan menjadi lebih efisien, serta meningkatkan peluang diagnosis dini.

# TUJUAN AI KAMI

## **Segmentasi Tumor**

- Menghasilkan area spesifik pada MRI yang menunjukkan lokasi tumor.
- Memberikan overlay mask untuk membantu visualisasi.

# Steps

## Dataset Preparation

Dataset dipersiapkan dengan diklasifikasi.

## Data Preprocessing

Data diproses sebelum dijadikan input training agar lebih merepresentasikan output yang dimau.

## Model Training

Model dilatih dengan data yang sudah diproses.

## Implementation

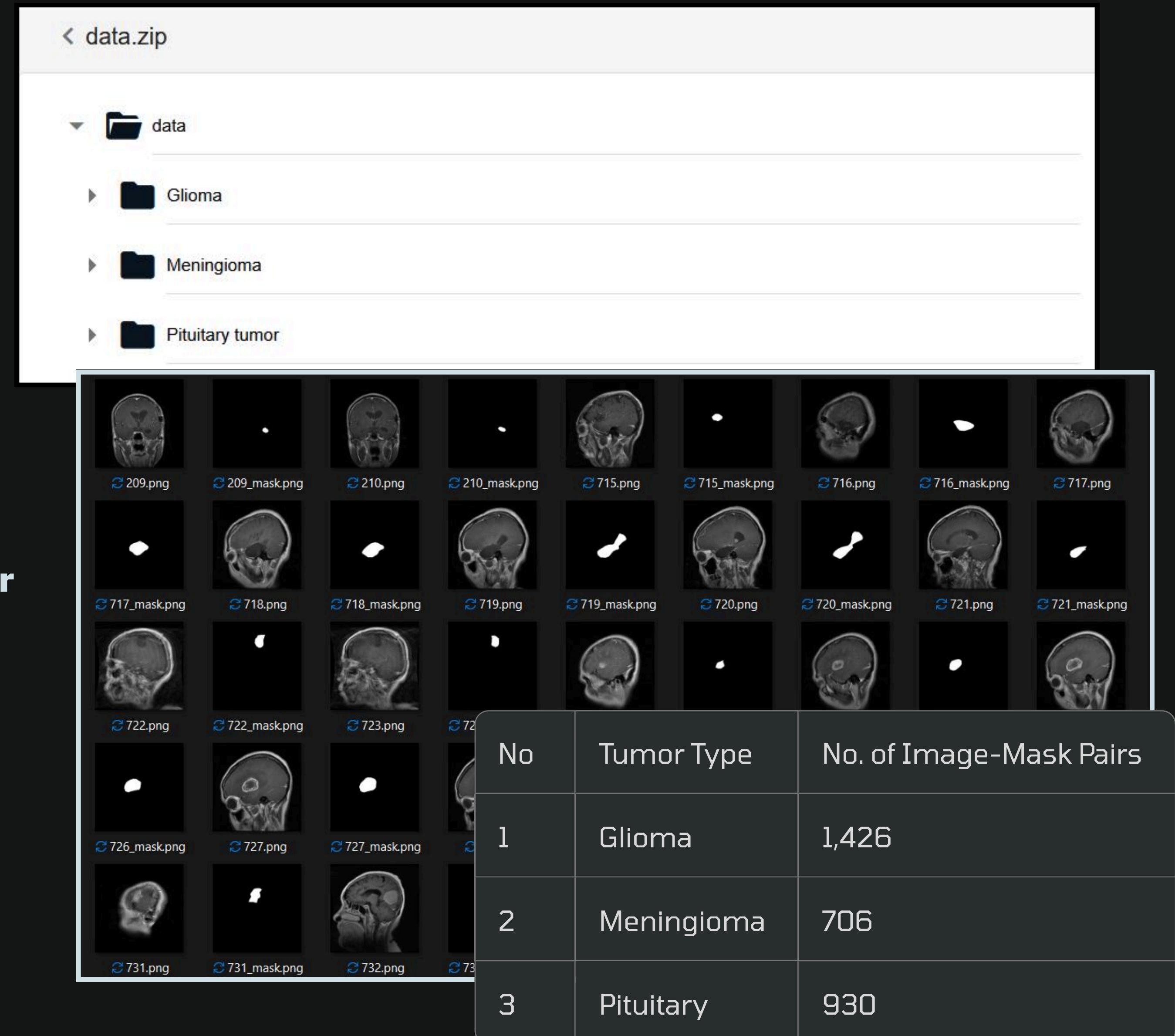
Implementasi model ke website dengan python flask.

# Science Data Bank

# DATASET

Dataset dari Science Data Bank (ScienceDB) yang merupakan kumpulan hasil scan MRI otak dari berbagai tampak. Dataset ini berisi **3.064 gambar MRI otak**, yang di mana setiapnya sudah diklasifikasikan dengan jenis tumornya, beserta dengan mask untuk tumornya.

Jun Cheng. Brain Tumor Dataset[DS/OL]. V1. Science Data Bank, 2022[2025-12-03]. <https://doi.org/10.57760/sciencedb.06290>. DOI:10.57760/sciencedb.06290.



# Dataset

**3.064** **Total Gambar MRI**

Dari dataset ScienceDB: [https://doi.org/10.57760/  
sciencedb.06290](https://doi.org/10.57760/sciencedb.06290)

**512x512** **Image Size**

Ukuran pixel image yang tersedia pada dataset

**233** **Patient Sources**

Jumlah pasien yang direpresentasikan dataset

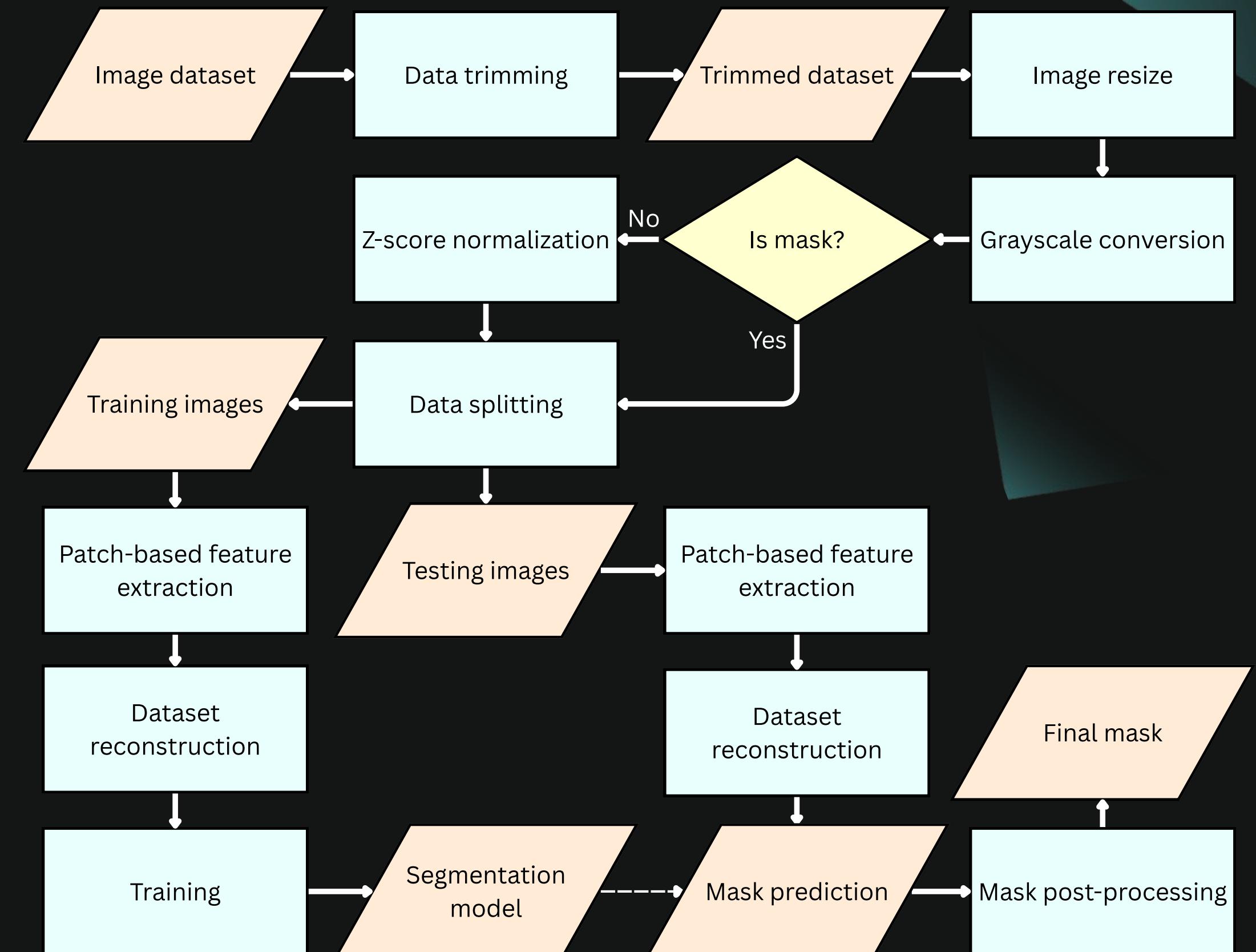
**2005-2010** **Year Gathered**

Tahun gambar-gambar MRI ini dikumpulkan

The images were acquired at Nanfang Hospital, Guangzhou, China,  
and General Hospital, Tianjin Medical University, China

# Outline

1. Image Processing
2. Split Dataset
3. Feature Extraction
4. Dataset Reconstruction
5. Training Segmentation
6. Evaluation
7. Post-processing
8. Application Development

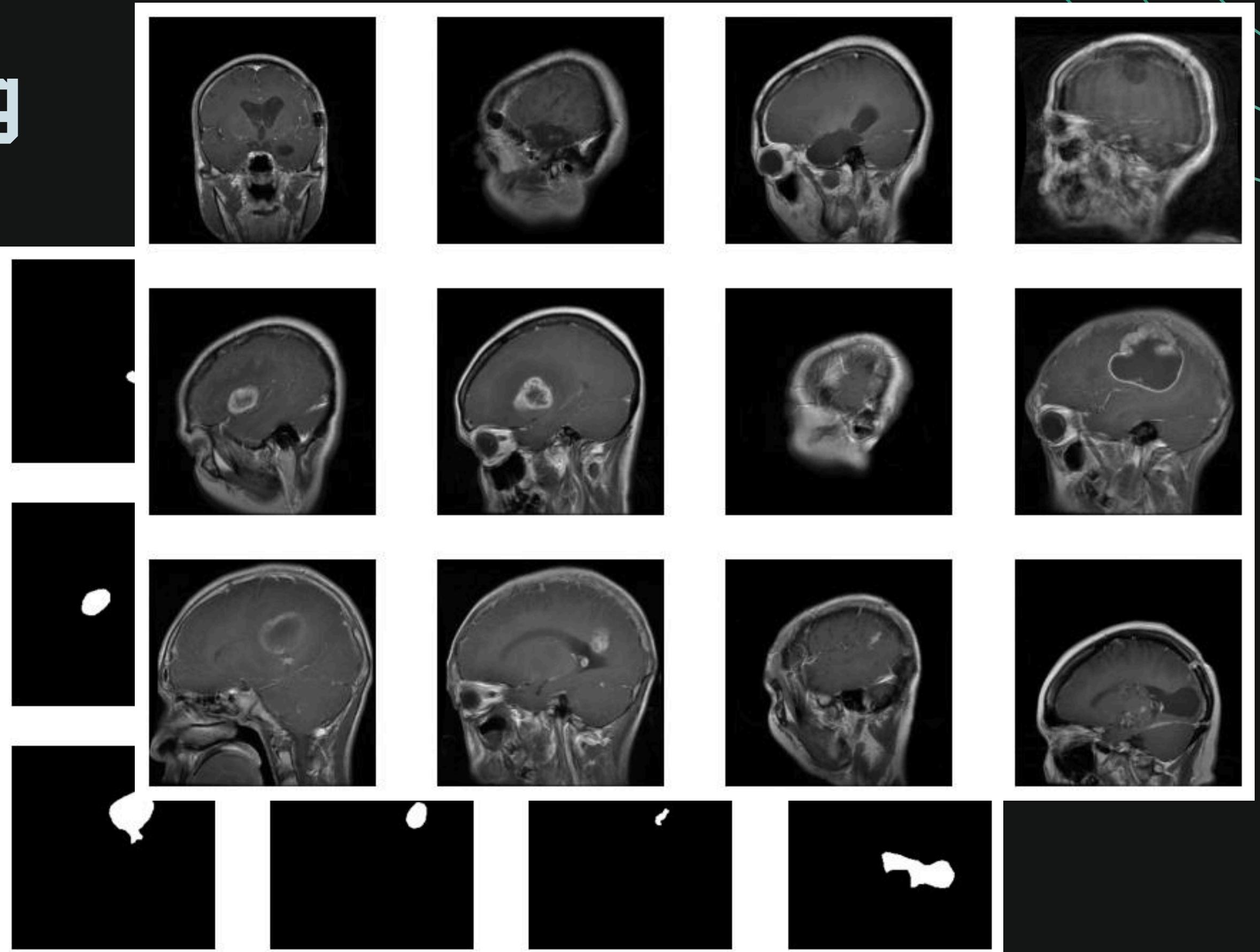


# 1. Image Processing

Image MRI diproses terlebih dahulu melalui:

- konversi ke grayscale,
- resize ke ukuran yang sama (256x256), dan
- normalisasi Z-score (kecuali mask).

Langkah ini memastikan setiap gambar memiliki format dan distribusi nilai piksel yang konsisten sebelum masuk ke tahap ekstraksi fitur.



## 2. Split Dataset

Dataset dibagi menjadi

- **70% train,**
- **15% validation, dan**
- **15% test.**

Setiap sampel terdiri dari image MRI, mask segmentasi, dan label jenis tumor (glioma, meningioma, pituitary). Oleh karena itu, model dapat belajar dari struktur tumor dan untuk pengembangan lebih lanjut kategorinya juga dapat dipakai secara bersamaan.

Total Utilized Dataset: 600 images  
Training set: 420 images  
Validation set: 90 images  
Test set: 90 images

Training set: (420, 256, 256), (420, 3), (420, 256, 256)  
Validation set: (90, 256, 256), (90, 3), (90, 256, 256)  
Testing set: (90, 256, 256), (90, 3), (90, 256, 256)

Images      Label      Mask

## 3. Feature Extraction

### Patch-Based Feature Extraction

Image dibagi menjadi patch kecil, kemudian diekstraksi fitur tekstur dan intensitas menggunakan intensity features dan GLCM. Ekstraksi patch ini membantu menangkap detail lokal untuk kebutuhan segmentasi.

Patch size: 7px

Image size: 256x256px

Total pixels in an image: 65.536px

Total features in a single image: 458.752 features

Features per pixel:

- Mean of intensity
- Variance of intensity
- GLCM Contrast
- GLCM Correlation
- GLCM Energy
- GLCM Homogeneity
- GLCM Entropy

```
feature_vector = [  
    mean_intensity,  
    var_intensity,  
    glcm_feats['contrast'],  
    glcm_feats['correlation'],  
    glcm_feats['energy'],  
    glcm_feats['homogeneity'],  
    glcm_feats['entropy']]
```

# Feature Extraction Process

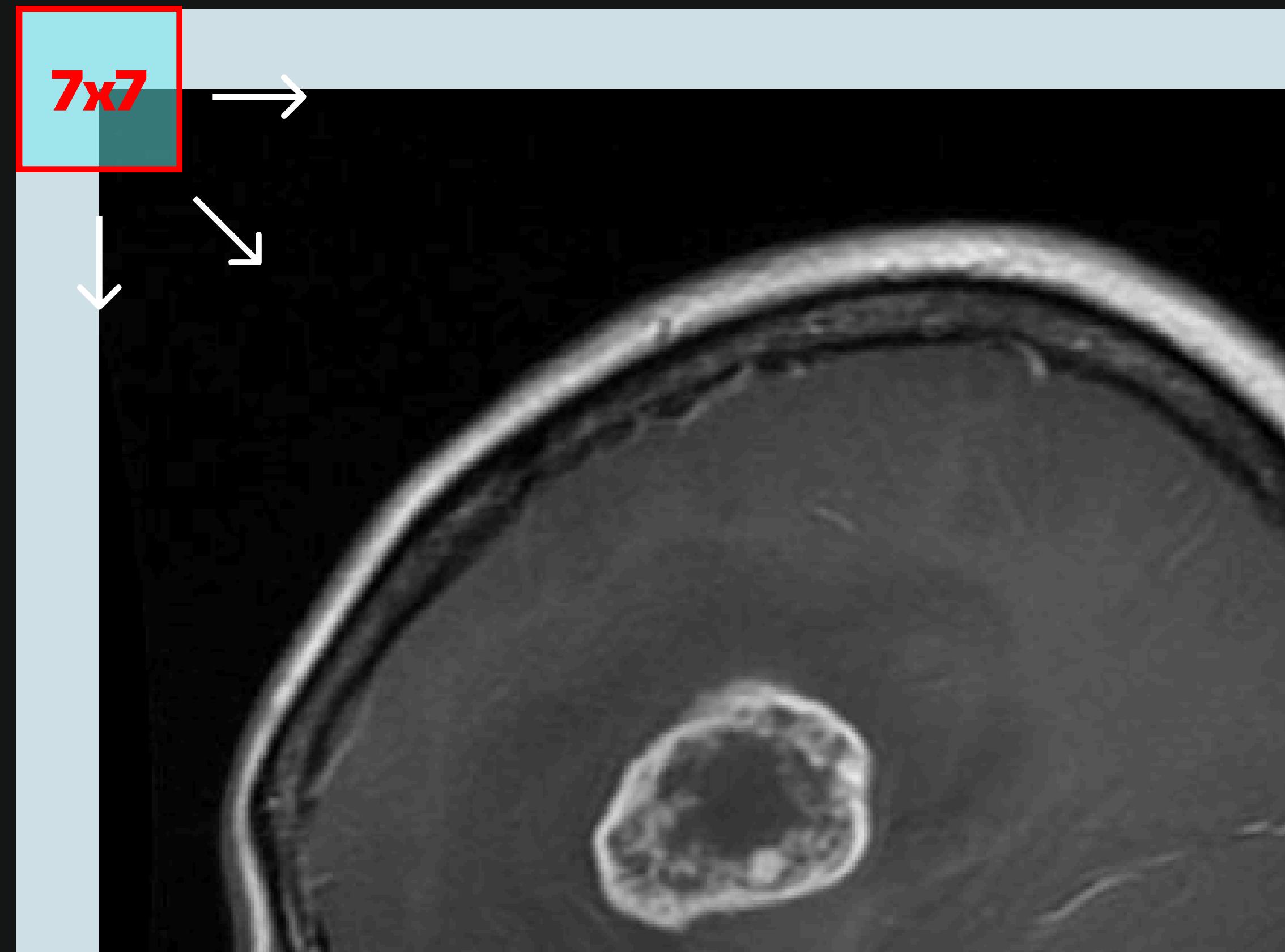
Raw Image



Padded Image



Patch-based extraction



## 4. Dataset Reconstruction

Patch beserta fitur-fitur yang telah diekstraksi disusun kembali menjadi dataset baru. Setiap image direpresentasikan sebagai kumpulan patch berfitur lengkap yang siap digunakan untuk pelatihan model segmentasi.

Dilakukan untuk Training, Validation, dan Testing set.

### Dataset Reconstruction

```
1 results = Parallel(n_jobs=-1, backend="loky")(| # Multi-processing for speed-up
2     delayed(build_dataset_from_image)(img, mask)
3     for img, mask in tqdm(zip(X_train, M_train), total=len(X_train), desc="Building training d
4 )
5
6 # Unpack results
7 X_all = [r[0] for r in results]
8 M_all = [r[1] for r in results]
9
10 # Final dataset
11 X_train = np.vstack(X_all)
12 M_train = np.hstack(M_all)
13
14 print(f'Final training dataset shape: {X_train.shape}, {M_train.shape}')
15
16 np.savez('train_varied_data.npz', X_train=X_train, M_train=M_train)
```

14]

.. Building training dataset: 0%| 0/420 [00:00<?, ?it/s]  
Building training dataset: 100%|██████████| 420/420 [23:29<00:00, 3.36s/it]  
Final training dataset shape: (27525120, 7), (27525120, )

## 5. Segmentation Training

Model segmentasi dilatih menggunakan Random Forest, memanfaatkan fitur patch untuk membedakan area tumor dan non-tumor secara otomatis.

```
1 rf_seg.fit(X_train, M_train)
```

```
[Parallel(n_jobs=-1)]: Using backend ThreadingBackend with 16 concurrent workers.  
[Parallel(n_jobs=-1)]: Done 18 tasks | elapsed: 7.8min  
[Parallel(n_jobs=-1)]: Done 125 out of 125 | elapsed: 31.6min finished
```

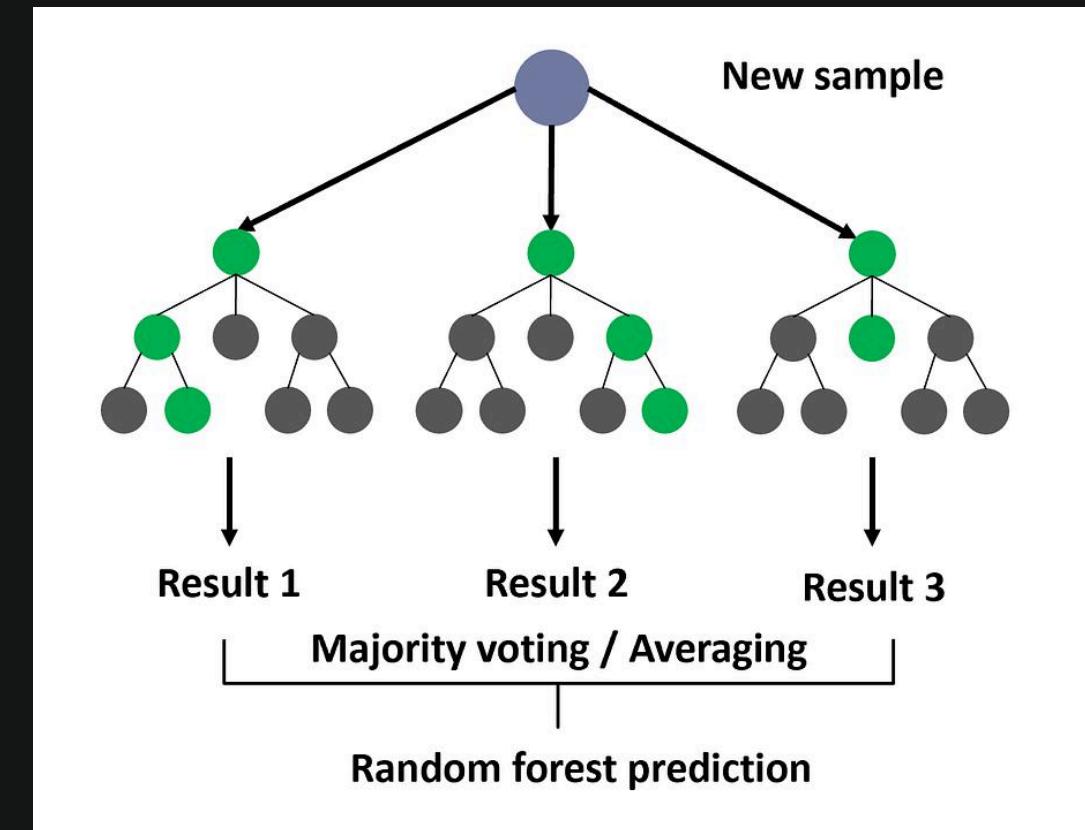
```
RandomForestClassifier  
RandomForestClassifier(class_weight='balanced', max_depth=40, max_samples=0.5,  
min_samples_leaf=2, n_estimators=125, n_jobs=-1,  
random_state=42, verbose=1)
```

# AI Model

Menggunakan classical machine learning.  
Kami menggunakan Random Forest dengan  
banyak tree 125 dan max depth 40.

```
RandomForestClassifier
RandomForestClassifier(class_weight='balanced', max_depth=40, max_samples=0.5,
min_samples_leaf=2, n_estimators=125, n_jobs=-1,
random_state=42, verbose=1)
```

```
1 rf_seg = RandomForestClassifier(
2     n_estimators=125,
3     max_depth=40,
4     class_weight='balanced',
5     min_samples_leaf=2,
6     max_samples=0.5,
7     random_state=42,
8     verbose=1,
9     n_jobs=-1
10 )
```



## 6. Evaluation

Model dievaluasi menggunakan metrik accuracy, dan Dice Coefficient/F1-score untuk menilai kualitas hasil segmentasi.

```
1 M_val_pred = rf_seg.predict(X_val)
2
3 print("Pixel Accuracy:", accuracy_score(M_val, M_val_pred))
4 print("Dice/F1 Score:", f1_score(M_val, M_val_pred, average='weighted'))
```

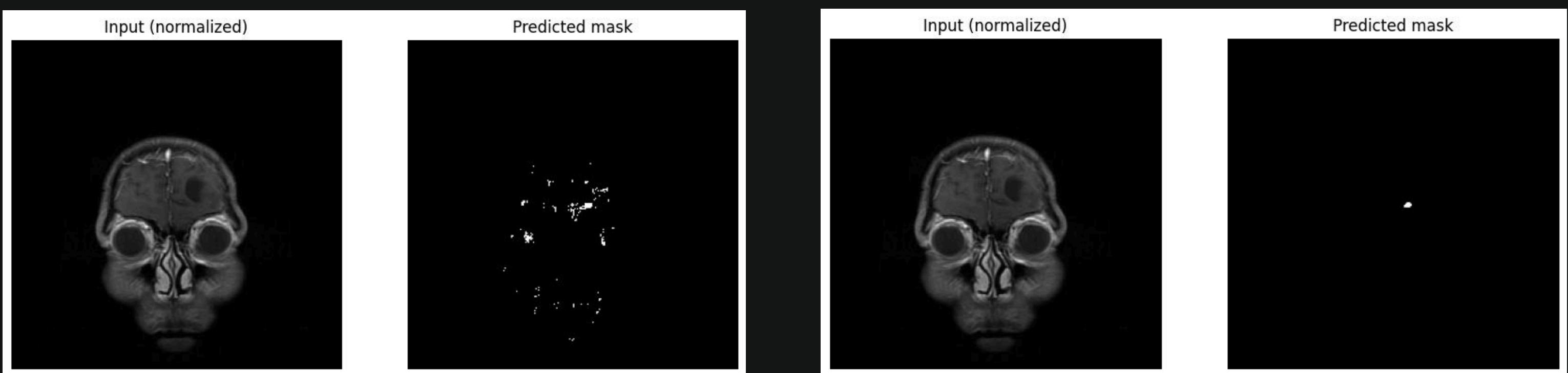
```
[Parallel(n_jobs=16)]: Using backend ThreadingBackend with 16 concurrent workers.
[Parallel(n_jobs=16)]: Done 18 tasks      | elapsed:    8.6s
[Parallel(n_jobs=16)]: Done 125 out of 125 | elapsed:   43.2s finished
Pixel Accuracy: 0.951404317220052
Dice/F1 Score: 0.9526744105722116
```

## 7. Post-Processing

- Probability map
- Gaussian Smoothing
- Binary mask conversion
- Connected Components (select the largest detected region)

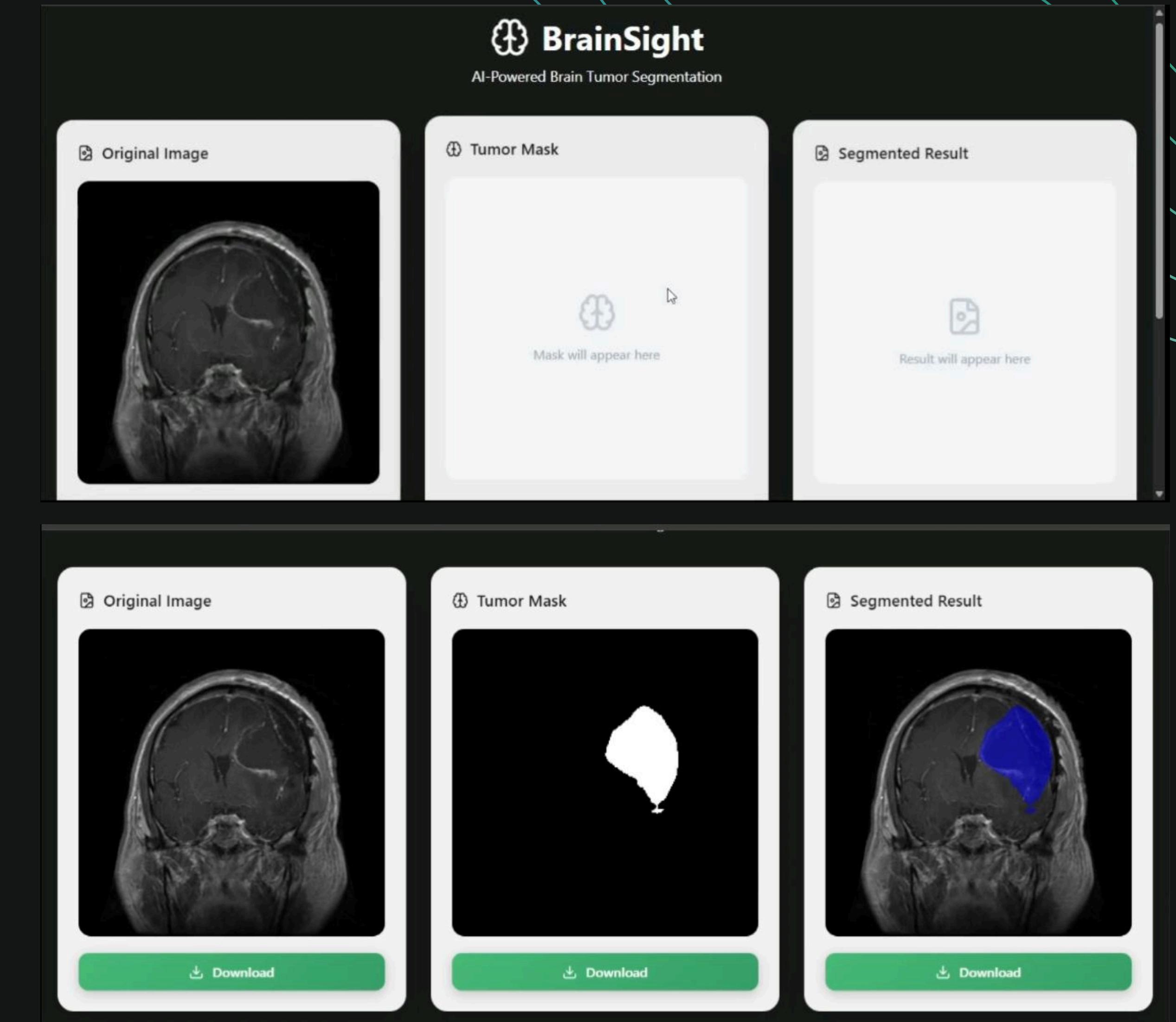
```
# Predict labels
proba_map = rf_model.predict_proba(features)[:,1].reshape(h, w)
smooth_map = cv2.GaussianBlur(proba_map, (5,5), 2) # smoothen probability map
binary_mask = (smooth_map > 0.5).astype(np.uint8)

num_labels, labels, stats, centroids = cv2.connectedComponentsWithStats(binary_mask)
if stats.shape[0] > 1:
    largest = 1 + np.argmax(stats[1:, cv2.CC_STAT_AREA])
    final_mask = (labels == largest).astype(np.uint8)
else:
    final_mask = np.zeros((h, w), dtype=np.uint8)
```



## 8. Deploy

Model akhir dapat digunakan tenaga medis untuk mengunggah MRI dan mendapatkan hasil segmentasi tumor secara otomatis.



# TRAINING RESULTS

## Metrics

**95,14% Accuracy**

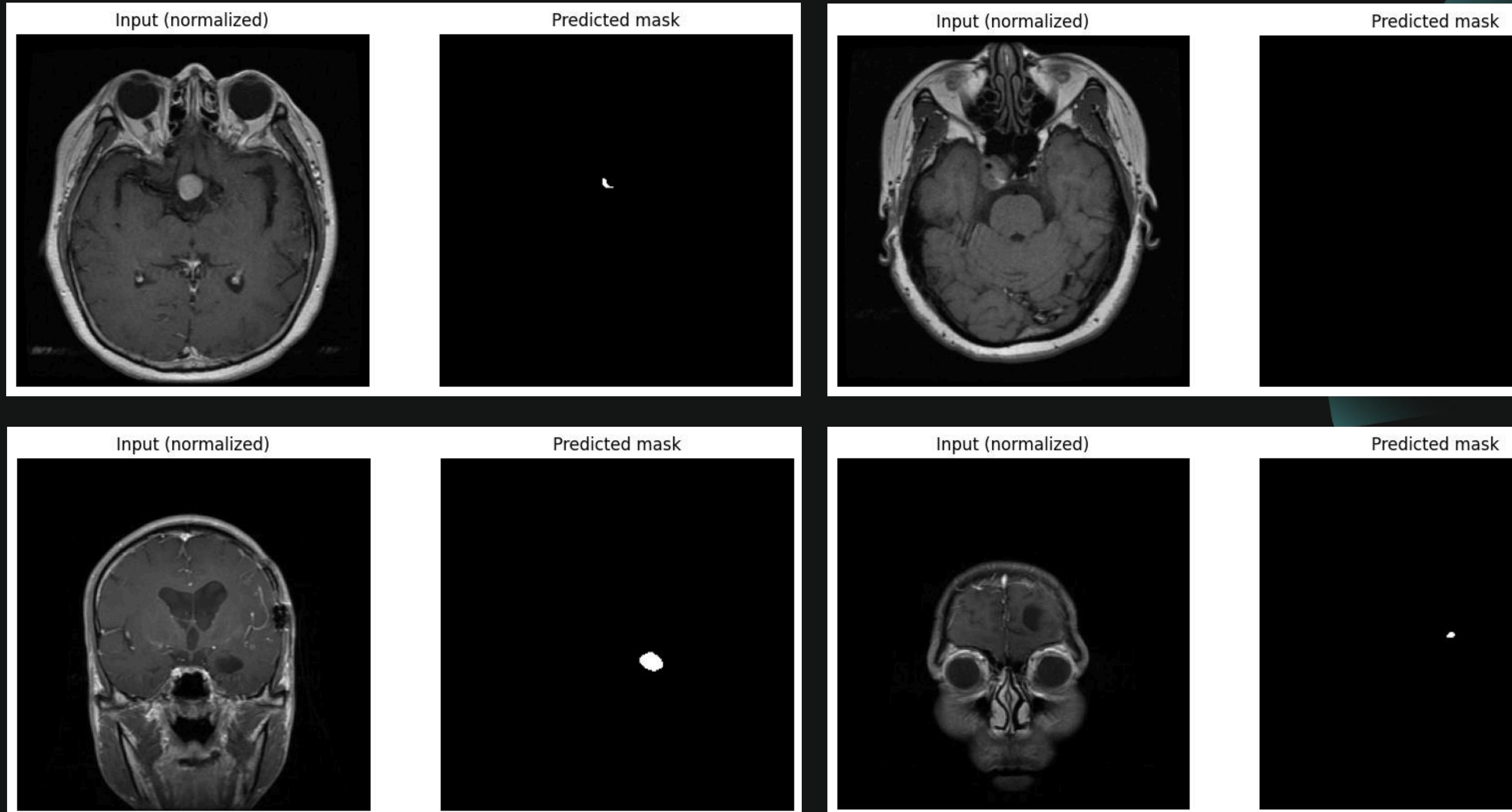
Akurasi segmentasi pada test set

**95,26% Dice/F1-Score**

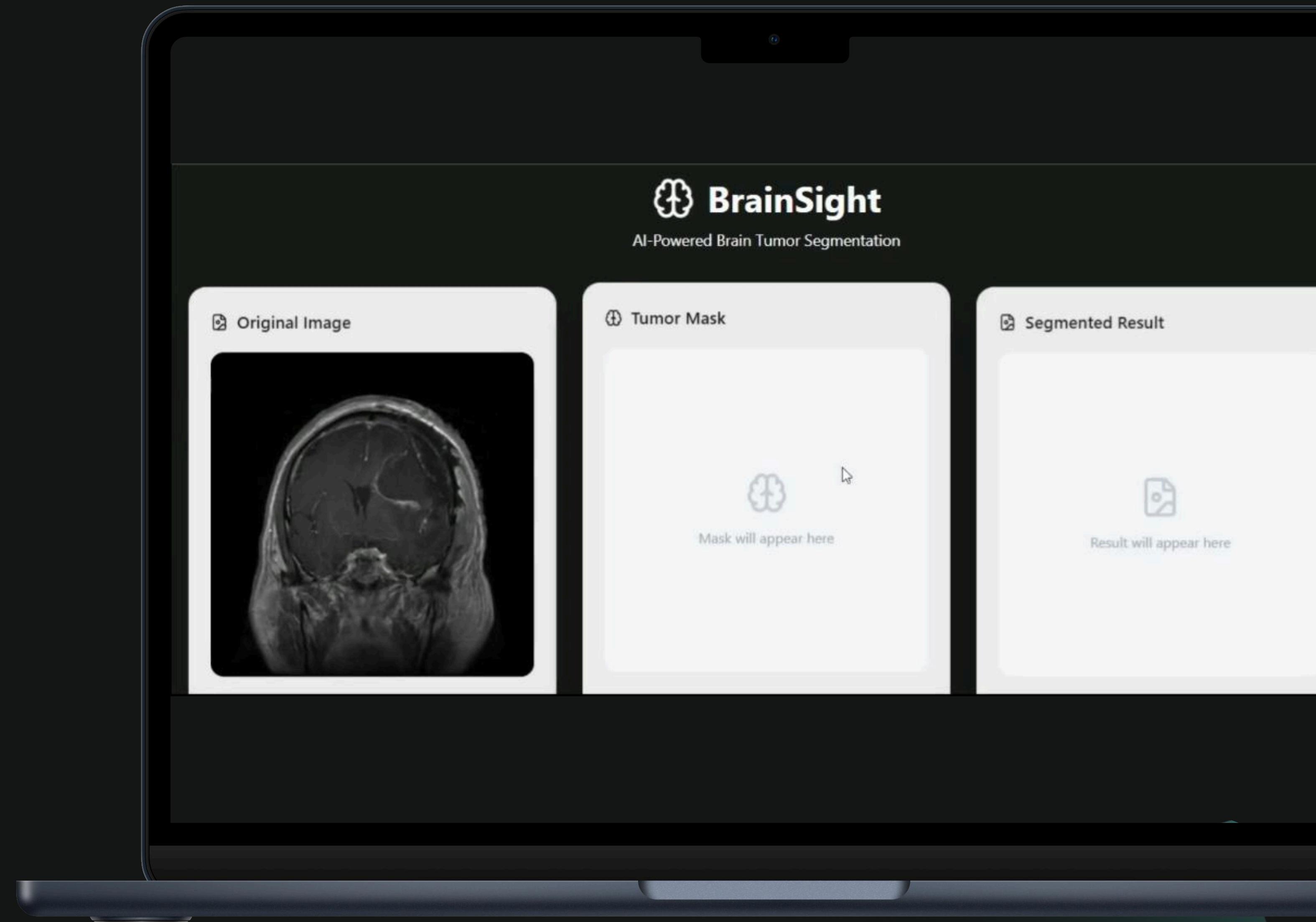
Dice coefficient/F1-score pada test set

```
Pixel Accuracy: 0.951404317220052
Dice/F1 Score: 0.9526744105722116
```

# Prediction Examples



# Implementasi Model



localhost:5173

AI-Powered Brain Tumor Segmentation

Upload Brain MRI Image

Drag and drop or click to browse

Supports: JPG, PNG, DICOM

© 2025 Brain Tumor Segmentation System  
For research and educational purposes only

11:12  
29/12/2025

# Contribution Statement

Albertus  
Ebert  
Chandrajaya

## Merumuskan Tema

Mendiskusikan dan memikirkan tema yang memenuhi kriteria final project

## Kontribusi Laporan Akhir

- Bagian abstract, introduction, related works, conclusion

## Video Akhir

- Berperan dalam mempresentasikan materi
- Merekam video di Zoom

## Kontribusi Git

Merapikan dokumentasi pada GitHub

## Menentukan Dataset

Membantu dalam mencari dan menentukan dataset yang pantas digunakan

## Membuat Slide Presentasi

Membantu pembuatan slide presentasi untuk presentasi akhir

# Contribution Statement

## Kent Amadeo Timotheus

### Merumuskan Tema

Mendiskusikan dan memikirkan tema yang memenuhi kriteria final project

### Kontribusi Laporan Akhir

- Bagian metodologi dan diskusi
- Membantu merapikan laporan

### Kontribusi Git

Melakukan push dan merge selama preprocessing dan pembuatan model

### Manajemen Kelompok

Membantu membagi tugas dan merencanakan timeline projek

### Menentukan Dataset

Membantu dalam mencari dan menentukan dataset yang pantas digunakan

### Pembuatan Model AI

- Riset model AI yang cocok
- Koding pre-processing, feature extraction, training, evaluasi performa, dan post-processing

### Membuat Slide Presentasi

Memantau dan membantu pembuatan slide presentasi untuk presentasi akhir

### Video Akhir

- Berperan dalam mempresentasikan materi
- Merekam video di Zoom

# Contribution Statement

Theodore  
Zachary

## Merumuskan Tema

Mendiskusikan dan memikirkan tema yang memenuhi kriteria final project

## Kontribusi Laporan Akhir

Bagian implementation and results

## Kontribusi Git

Melakukan push selama pembuatan implementasi aplikasi

## Pembuatan Web

Melakukan development web, koding untuk frontend dan backend

## Membuat Slide Presentasi

Memantau dan membantu pembuatan slide presentasi untuk presentasi akhir

## Video Akhir

- Berperan dalam mempresentasikan materi
- Merekam video di Zoom

# THANK YOU

**Resources:**

GitHub: [github.com/Garchompz/cv-brain-tumor-classification-segmentation](https://github.com/Garchompz/cv-brain-tumor-classification-segmentation)