Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Bölümü

Derin Öğrenme Kullanarak MR Görüntülerinden Beyin Tümörü Sınıflandırması

Derin Öğrenme

22822606
Oğuzhan Panatlı
Doktora Öğrencisi

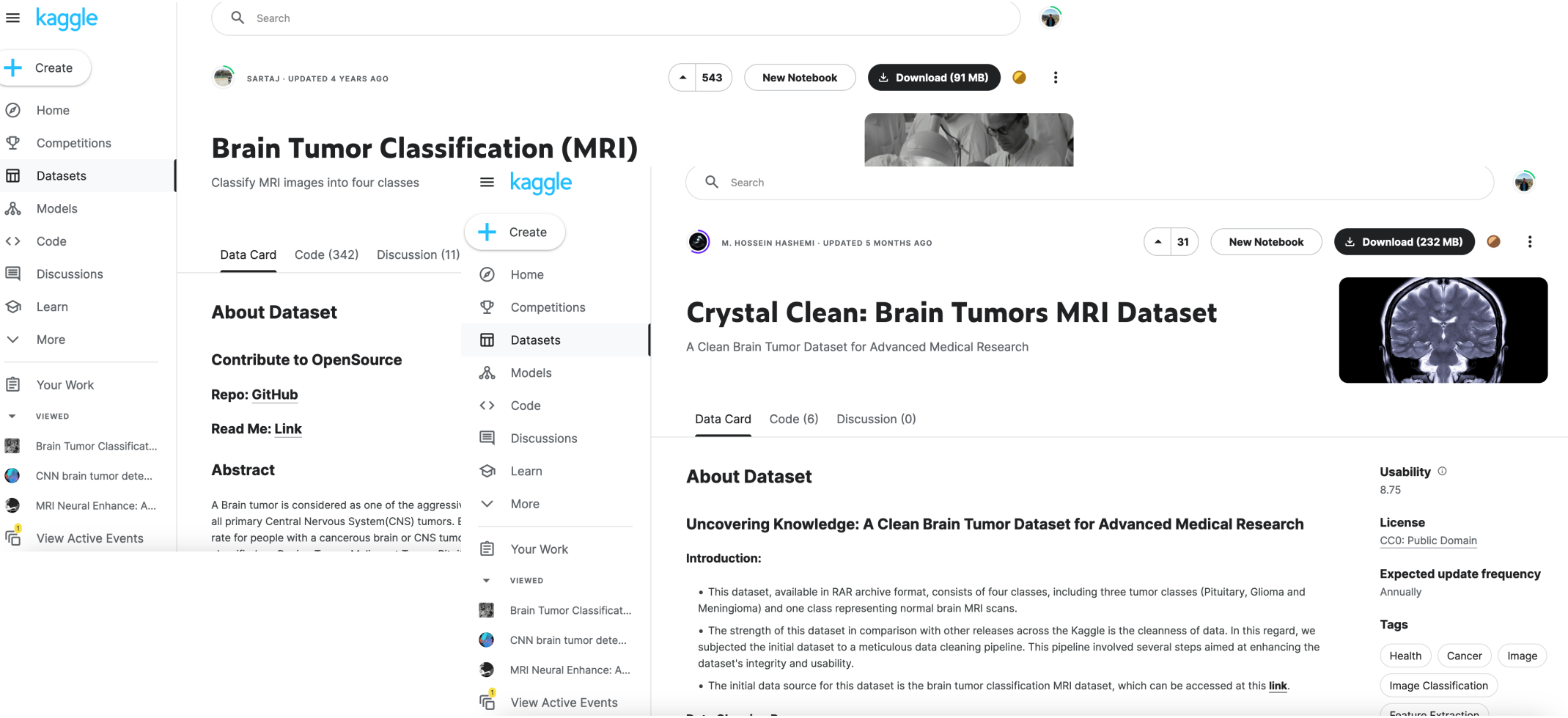
Doç. Dr. Mehmet Serdar Güzel
Dersin Öğretim Üyesi

Aralık 2023

İçindekiler

- Giriş
  - Veri Setine Genel Bakış
  - Veri Setinin İncelenmesi
- Yöntem
  - Veri Ön İşleme
  - Derin Öğrenme Modelinin (CNN) Uygulanması
  - Performans Metrikleri
- Sonuçların Değerlendirmesi
  - CNN Modelinin Sonuçlarının Değerlendirilmesi ve Yorumlanması
- Sonuç

Veri Setine Genel Bakış



https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri

https://www.kaggle.com/datasets/mohammadhossein77/brain-tumors-dataset

Veri Setine Genel Bakış

- Veri setinde **dört** sınıf bulunmaktadır: Normal, Pituitary, Glioma ve Meningioma.
  - **Normal -** Herhangi bir tümör belirtisi göstermeyen beyin görüntüsü
  - **Pituitary -** Hipofiz bezinde büyüyen bir tümör. Hipofiz tümörleri iyi huylu veya kötü huylu olabilir.
  - **Glioma -** Glial hücrelerde büyüyen bir tümör.
  - **Meningioma -** Beyni ve omuriliği çevreleyen zarlar olan meninkslerde büyüyen bir tümör.

https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri

https://www.kaggle.com/datasets/mohammadhossein77/brain-tumors-dataset

Veri Setine Genel Bakış

- Veri seti '**Brain Tumor Classification (MRI)'** kullanarak oluşturulmuştur.

- Veri Temizleme (Data Cleaning):
  - Tekrarlanan örneklerin kaldırılması (Removal of duplicate samples)
  - Yanlış etiketlenmiş görsellerin düzeltilmesi (Correction of mislabeled images)
  - Resim yeniden boyutlandırma (Image resizing): Tüm resimler **(224,224)** boyutuna boyutlandırılmıştır.

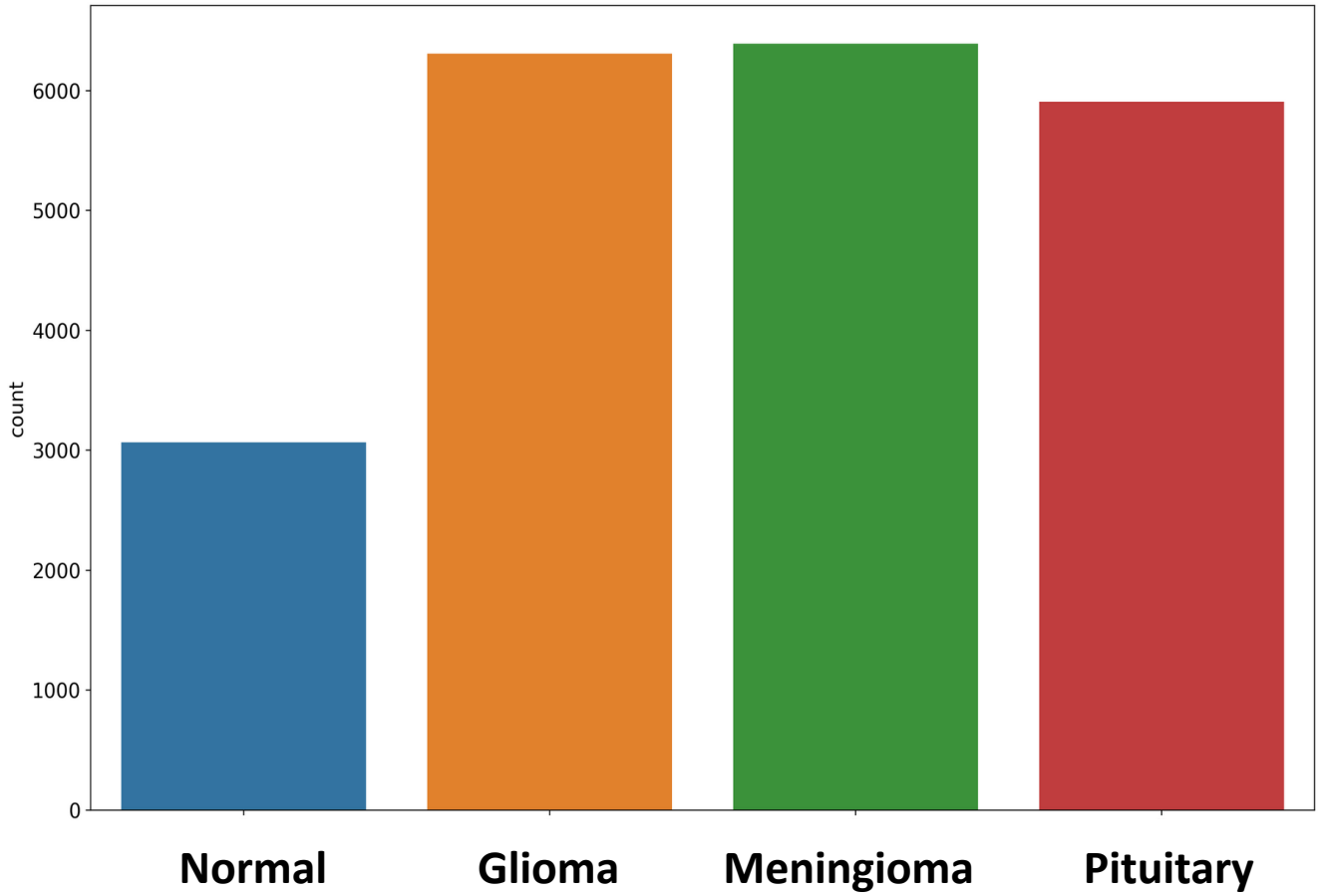
- Veri Artırma (Data Augmentation):
  - Tuz ve biber gürültüsü (Salt and pepper noise)
  - Histogram eşitleme (Histogram equalization) : Görüntülerdeki kontrastı ve ayrıntıları artırır.
  - Döndürme (Rotation)
  - Parlaklık ayarı (Brightness adjustment)
  - Yatay ve dikey çevirme (Horizontal and vertical flipping)

https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri

https://www.kaggle.com/datasets/mohammadhossein77/brain-tumors-dataset

Veri Setininin İncelenmesi

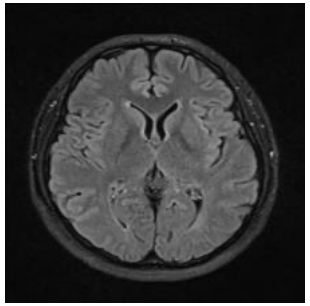
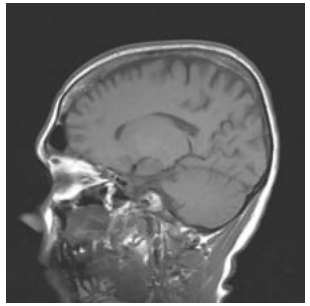
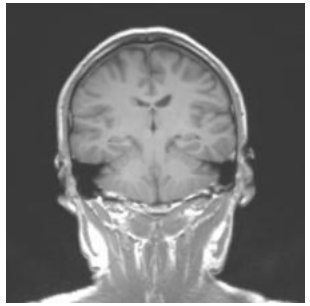
- Veri Setindeki gözlem sayısı (sample): **21672**
  - **Normal:** 3066
  - **Glioma:** 6307
  - **Meningioma** : 6391
  - **Pituitary:** 5908



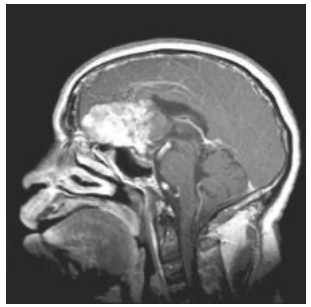
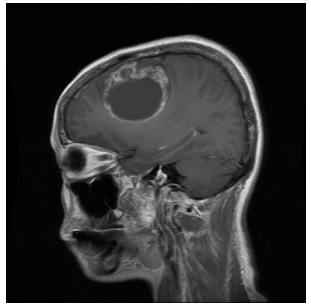
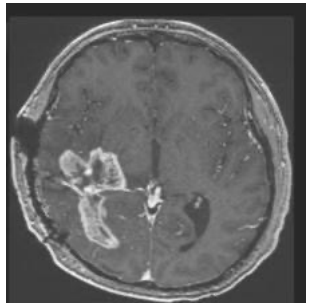
https://www.kaggle.com/datasets/mohammadhossein77/brain-tumors-dataset

Veri Setininin İncelenmesi

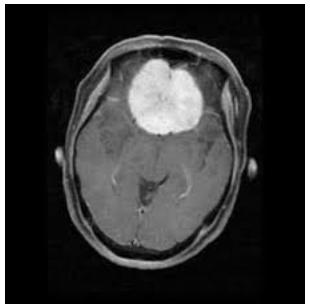
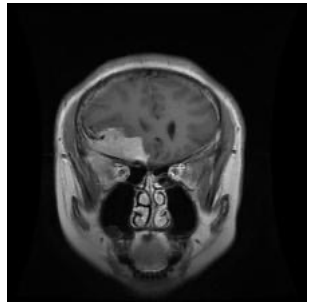
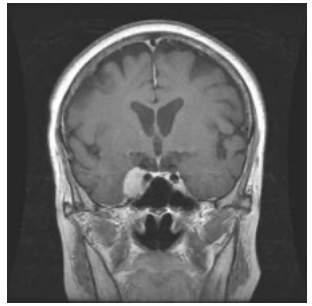
Normal

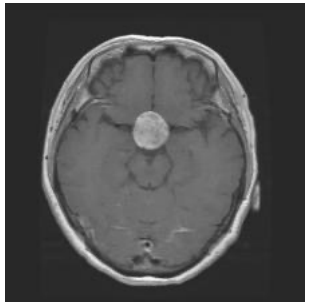
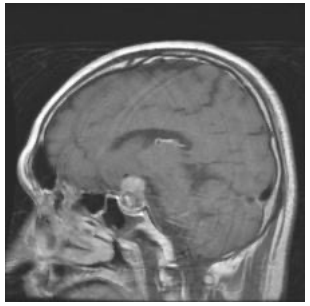
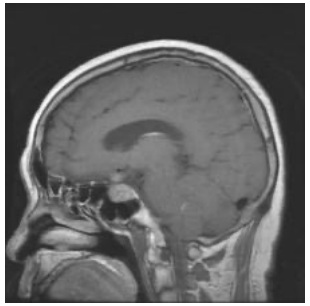
Glioma

Meningioma

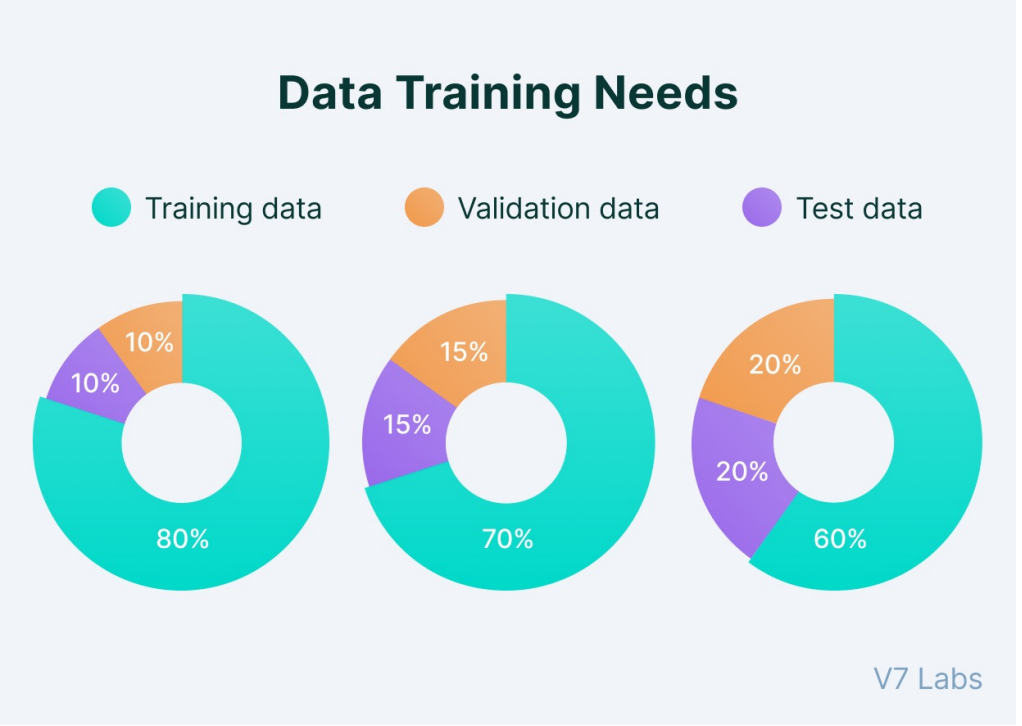
Pituitary

https://www.kaggle.com/datasets/mohammadhossein77/brain-tumors-dataset

Veri Ön İşleme

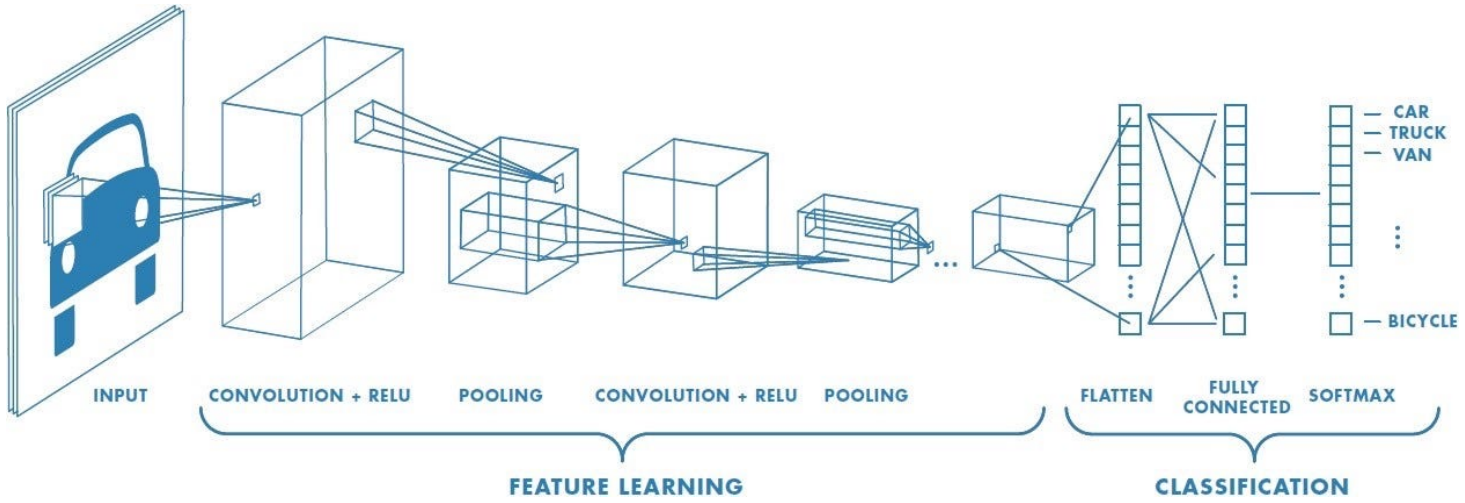
- Veri setinin eğitim, validasyon ve test olarak ayrılması (train-test split)
  - Eğitim Veri seti **~ %64,** Validasyon Veri Seti **~%16,** Test Veri seti **%20**
- Sınıfların sayısal verilere dönüştürülmesi
  - Normal, Glioma, Meningioma, Pituitary
- Görüntünün gri tonlamaya dönüştürülmesi (gray scale)
- Yeninden boyutlandırılması ve normalize edilmesi
  - Görüntü boyutu: **(150,150)**

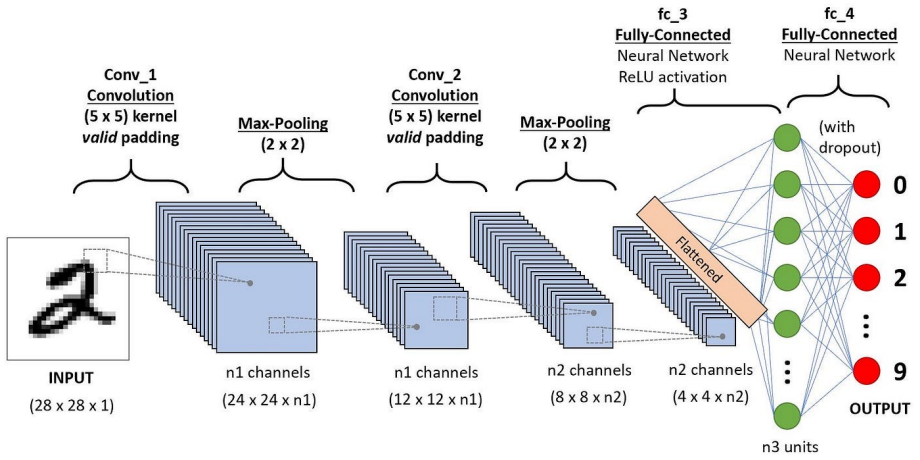


https://www.v7labs.com/blog/train-validation-test-set

Derin Öğrenme Modelinin (CNN) Uygulanması

- Convolutional Neural Network (CNN)





https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Derin Öğrenme Modelinin (CNN) Uygulanması

- Convolutional Neural Network (CNN)

```
# Training parameters
epochs = 20
batch_size = 150

print(f'Image shape: {image_shape}')
print(f'Epochs: {epochs}')
print(f'Batch size: {batch_size}')
```

```
Image shape: (150, 150, 1)
Epochs: 20
Batch size: 150
```

```
# Define the model architecture
model = models.Sequential()

# Convolutional layer 1
model.add(Conv2D(32, (4, 4), activation="relu", input_shape=image_shape))
model.add(MaxPooling2D(pool_size=(3, 3)))

# Convolutional layer 2
model.add(Conv2D(64, (3, 3), activation="relu"))
model.add(MaxPooling2D(pool_size=(3, 3)))

# Convolutional layer 3
model.add(Conv2D(128, (3, 3), activation="relu"))
model.add(MaxPooling2D(pool_size=(3, 3))) #

# Convolutional layer 4
model.add(Conv2D(128, (3, 3), activation="relu"))
model.add(Flatten())

# Full connect layers
model.add(Dense(512, activation="relu"))
model.add(Dropout(0.5, seed=10))
model.add(Dense(N_TYPES, activation="softmax"))

model.summary()

optimizer = legacy.Adam(learning_rate=0.001)

model.compile(optimizer=optimizer, loss='categorical_crossentropy', metrics= ['accuracy'])
```

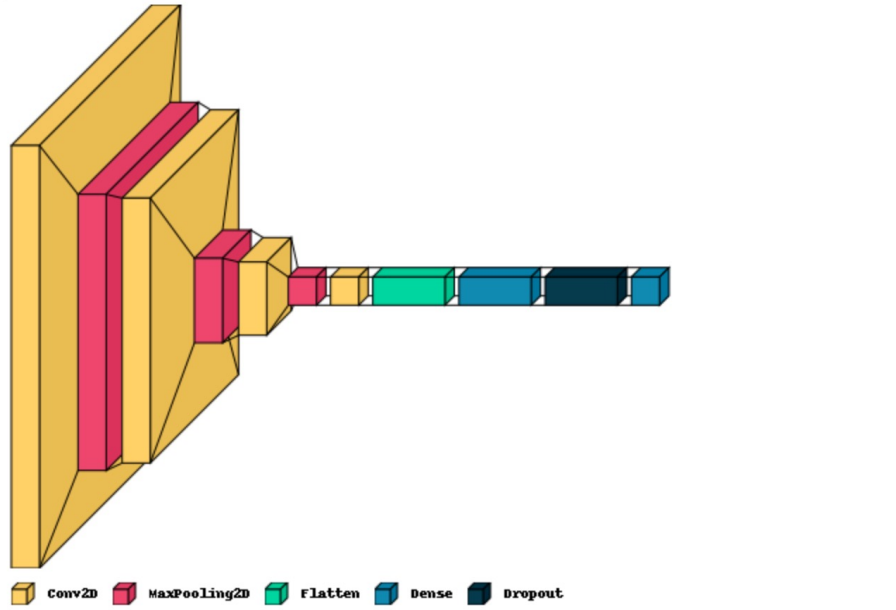
Derin Öğrenme Modelinin (CNN) Uygulanması

- Convolutional Neural Network (CNN)

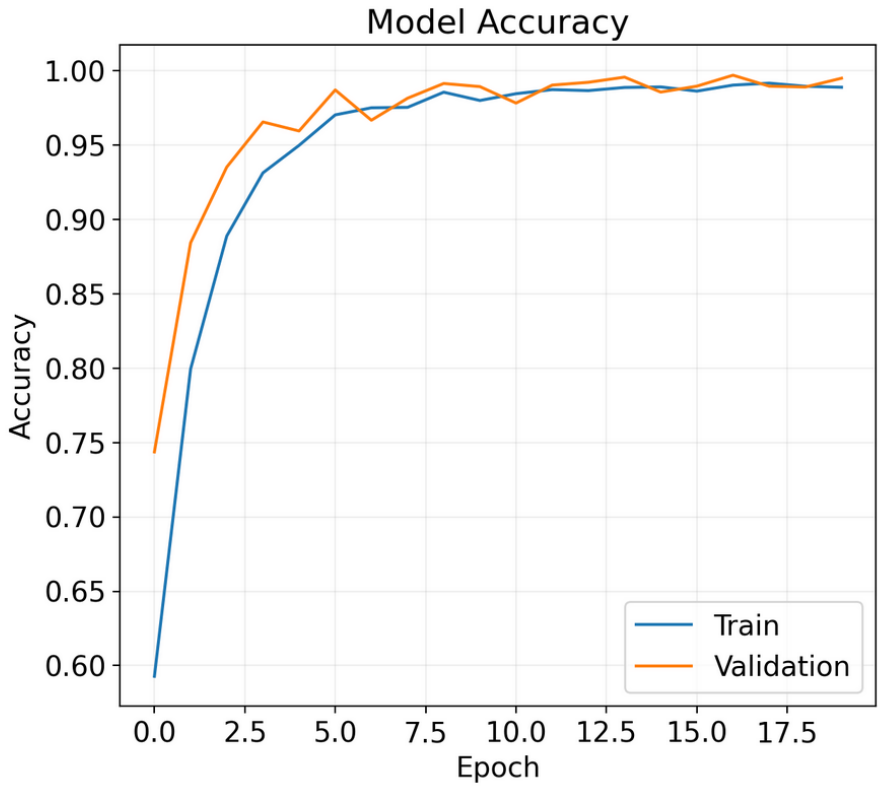
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 147, 147, 32)      544

 max_pooling2d (MaxPooling2  (None, 49, 49, 32)        0
 D)

 conv2d_1 (Conv2D)           (None, 47, 47, 64)        18496

 max_pooling2d_1 (MaxPoolin  (None, 15, 15, 64)        0
 g2D)

 conv2d_2 (Conv2D)           (None, 13, 13, 128)       73856

 max_pooling2d_2 (MaxPoolin  (None, 4, 4, 128)         0
 g2D)

 conv2d_3 (Conv2D)           (None, 2, 2, 128)         147584

 flatten (Flatten)           (None, 512)               0

 dense (Dense)               (None, 512)               262656

 dropout (Dropout)           (None, 512)               0

 dense_1 (Dense)             (None, 4)                 2052

=================================================================
Total params: 505188 (1.93 MB)
Trainable params: 505188 (1.93 MB)
Non-trainable params: 0 (0.00 Byte)
```

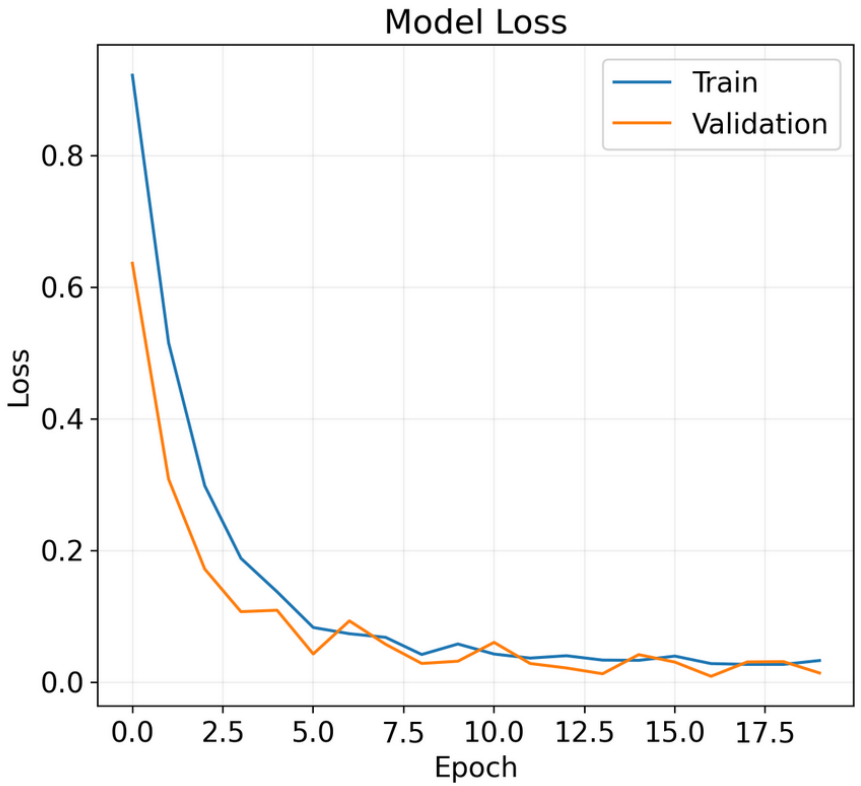
[14]:
```
from visualkeras import layered_view

# Visualize the model
layered_view(model, legend=True, max_xy=300)
```

[14…



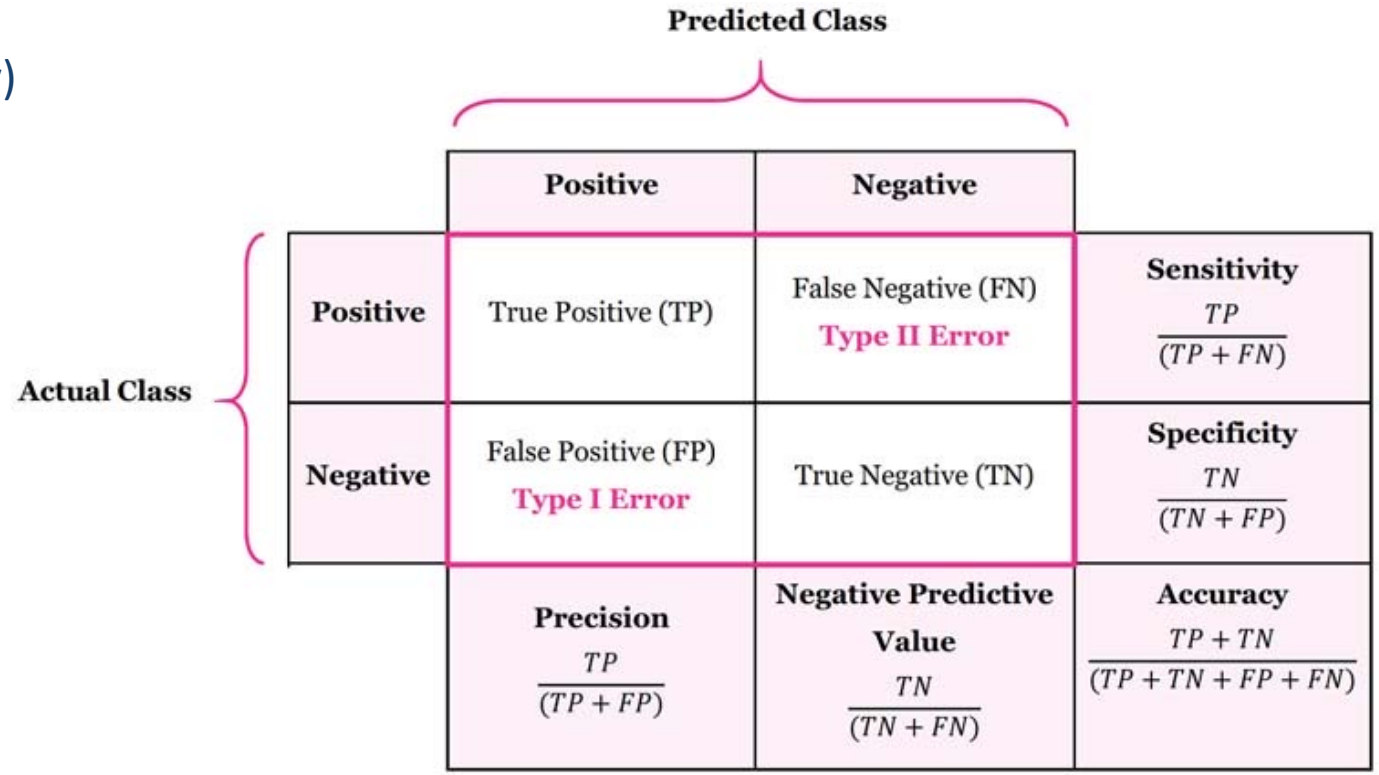
CNN Modelinin Sonuçlarının Değerlendirilmesi





Performans Metrikleri

- Model sonuçlarının değerlendirilmesinde kullanılan performans metrikleri
  - Accuracy
  - Precision
  - Recall (Sensitivity)
  - F1 Score



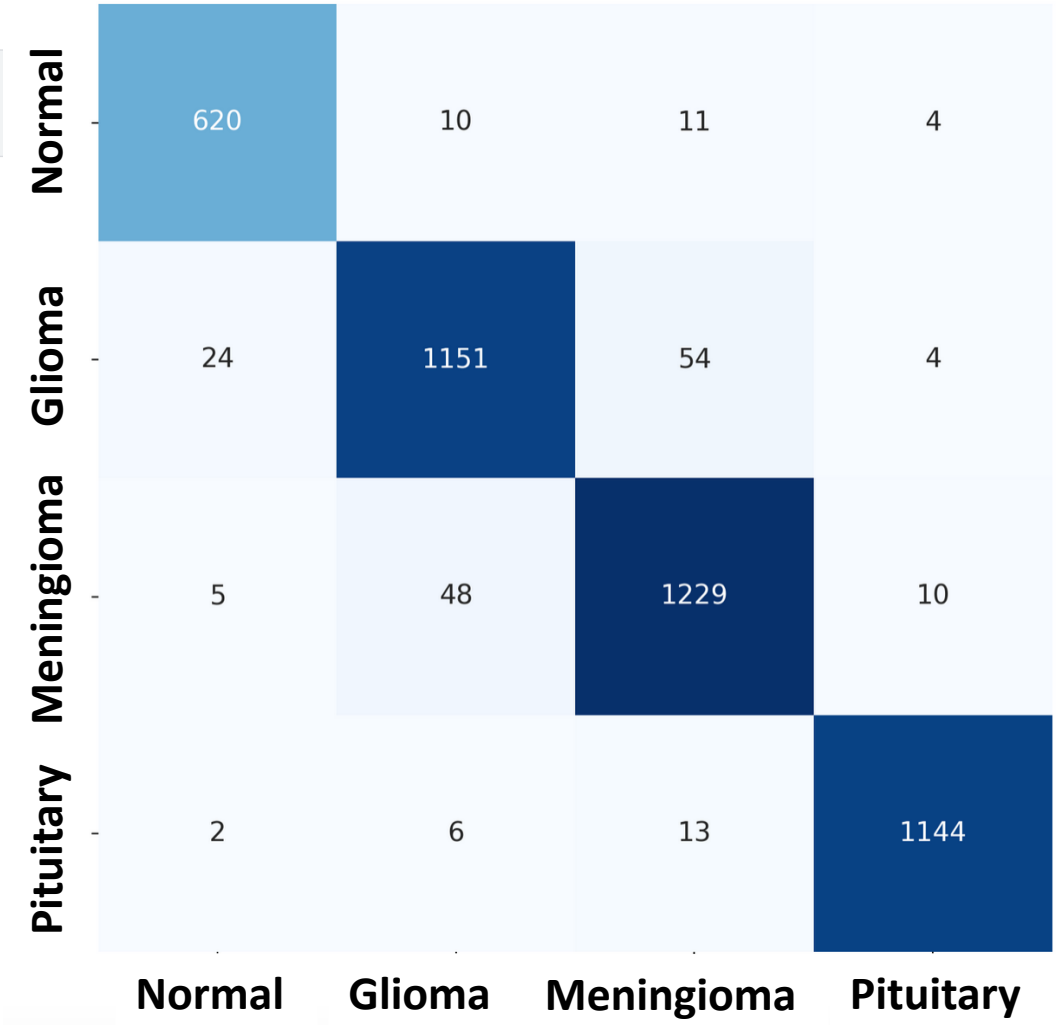
https://manisha-sirsat.blogspot.com/2019/04/confusion-matrix.html

CNN Modelinin Sonuçlarının Değerlendirilmesi

[149]:
```
test_score = model.evaluate(test_dataset)
print('Test Accuracy: {0:.3f} %'.format(test_score[1]*100))
```

```
136/136 [==============================] - 5s 18ms/step - loss: 0.2065 - accuracy: 0.9559
Test Accuracy: 95.594 %
```

```
Classification Report:
              precision    recall  f1-score   support

      Normal       0.95      0.96      0.96       645
      Glioma       0.95      0.93      0.94      1233
  Meningioma       0.94      0.95      0.95      1292
   Pituitary       0.98      0.98      0.98      1165

    accuracy                           0.96      4335
   macro avg       0.96      0.96      0.96      4335
weighted avg       0.96      0.96      0.96      4335
```



Sonuç

- Bu veri setinde CNN modeli uygulanarak sınıflandırma yapılmıştır.
- Bu veri setinde model ~%96 test doğruluğuna ulaşılmıştır.
- Daha yüksek doğrulukta bir model elde edebilmek adına ileri çalışmalarda;
  - Veri sayısı artıralarak (gerçek veriler, Generative Adverserial Network (GAN)) tekrar CNN uygulanabilir.
  - Hiperparametre optimizasyonu yapılarak tekrar model oluşturabilir.
  - VGG16, Resnet152,Densenet201, InceptionV3, Xception gibi farklı modeller kullanılabilir.

DİNLEDİĞİNİZ İÇİN TEŞEKKÜRLER