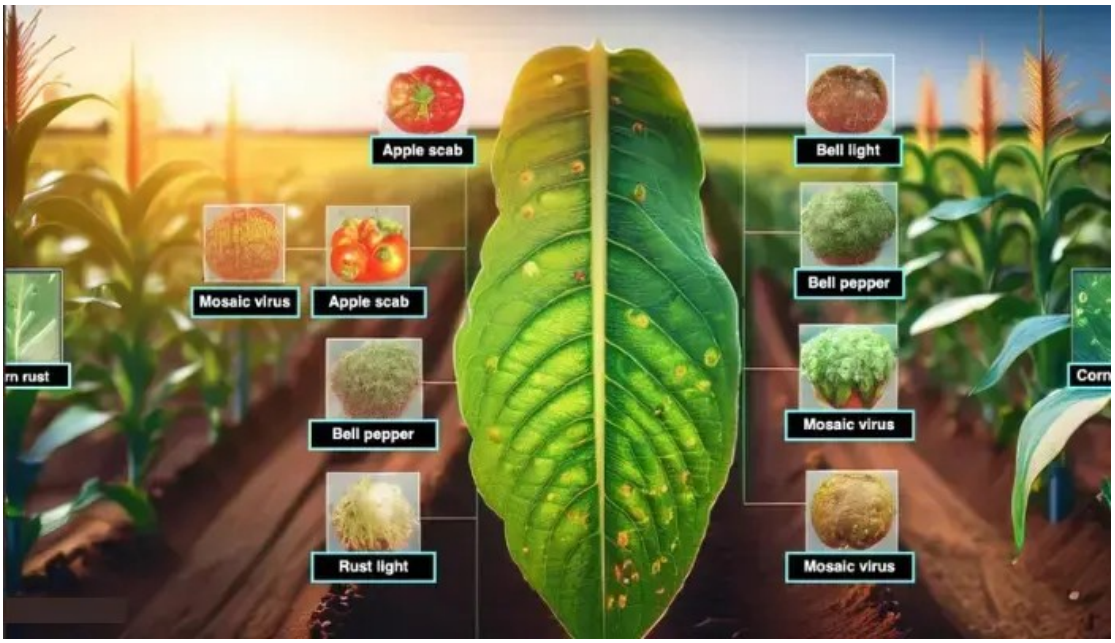


Bitki Hastalıkları Projesi

Genel Bilgilendirme Raporu

Ahmet efe Y-

27 Ocak 2026 Salı

Bitki hastalıkları projesi için geliştirmekte olduğum programın kabaca %65 i tamamlanmış durumdadır; bu projede ***Temelde hastalık tespiti-hasta/saglikli ayrımı*** *olmak üzere iki özellik bulunmakta bu özellikler için iki ayrı model bulunmakta bu modeller temel alt yapısında* ***CNN*** *yani* ***Convensional Neural Network'****leri kullanmaktadır* ***Hasta-Sağlıklı*** *ayrımı yapan model bizlere 0-1 arası bir değer verir bu değer* ***1 e ne kadar yakınsa ve 0 dan ne kadar uzak ise bitki o kadar sağlıklı dır*** *metriğine göre programdaki karar mekanizmaları sayesinde ölçülür ve*

0.5 ten büyük olan çıktılar sağlıklı ***(0.4) + 0*** *ve* ***(0.4) + 0,05*** *arası gelen değerler ise* ***Kısmen sağlıklı*** *yani* ***Half Healthy*** *kategorisinde değerlendirilir*
ve ***Hasta*** *yani* ***Diseased*** *olarak değerlendirilenler ise* ***(0.4) + 0*** *ve* ***(0.4) + 0,05*** *arasındaki değerlerinde altında olan çıktılardır;*

Model çıktılarının program tarafından yorumlanmasındaki metrikler bu şekilde dile getirilebilir ve teknik olarak açıklanabilir şimdi ise henüz eğitilmemiş olan Hastalık sınıflandırma modeline bir göz atalım;

BU modelde bitki hasta ise ve bu hastalık bilinmiyor ise gelen görseli X hastalık sayısınca tahmin eder ve X tane modele gösterilen hastalık türleri ile en uyumlu eşleşen sınıfa 0-1 arası en 1 e yakın değeri verir örneğin ***4*** *sınıflı bir yapı olsa çıktı şu şekilde olur*

[[0.05616~ , 0.03043~, 0.20510~, 0.50331]]

Şeklinde bir çıktı gelir burada aslında sınıflar şu şekildedir

{'X hastalığı': 0 , 'Y hastalığı': 1, 'Z hastalığı': 2, 'Q hastalığı': 3}

yani burada

X hastalığı : 0.05616~
Y hastalığı : 0.03043~
Z hastalığı : 0.20510~
Q hastalığı : 0.50331~

Şeklinde bir karar verilir ve ***Q hastalığı*** *tahminler yani* ***Prediction****'lar arasından en maksimum olanı olarak seçilir yani kısaca karar verilirken çok sınıflı yapılarda model bu şekilde her sınıfa* ***skorlar*** *biçer ve programda bunları metriklere göre analiz eder*

Bu projedeki 0-1 arası sınıflandırma yapan model ile çok sınıflı bir şekilde hastalık tespiti yapan modeli karıştırmayın ikisi farklı modeller ve şimdi ise 0-1 arası sınıflandırma yapan modelin girdi olarak neler aldığına bakalım;

Temel olarak bu model bizden girdi olarak 0-255 arası normalize edilmiş RGB uzay boyutlu görselin 0. axisine batch dimension ***Batch boyutu*** *eklenmiş ve 224x224 olarak yeniden boyutlandırılmış RGB renk uzayına sahip bir görsel ister görseller genel olarak farklı*

boyutlarda ve çoğunlukla BGR2 renk uzaylarında gelirler ayrıca 0-255 arası normalize edilmiş olarak gelmezler genelde görüntü matrisleri içinde INT yani tam sayılar bulunur 0-256 arası yani 0 dan 255 e kadar rakamlardan oluşan büyük matrislerdir ve modele bunları vermeden önce ilk başta belirttiğim işlermleri uygulamamız gerekir bu işlemleri pythonda istersek ***Tensorflow*** *un kendi fonksiyonları ile halledebilir veya istersekte harici olarak* ***opencv-python*** *ile halledebiliriz bu işleme* ***preprocess*** *yani* ***ön işleme*** *denir ön işleme esnasında modelin isteyeceği formatlara görüntü çevirilir ve* ***flow*** *yani* ***akış*** *esnasında fonksiyonlar aracılıgı ile dizinden çekilerek modele verilir ve besleme aşaması gerçekleşmiş olur ve modelin tahmin yapacağında ise bu aşama tekrarlanır ve tahmin edeceği görsel modele verilmeden bu aşamalardan geçirilir ve modele o şekilde veriliki model sapıtmasın saçma tahminler yapmasın veri tutarlı olsun.*

Bu aşamalarıda temel olarak anlattığıma göre artık programa geçebilirim;

Programda temel olarak model ile alakalı bir özgürlük mevcut; bahsettiğim şey şu; Yukarıda bahsettiğim 2 tane model ile çalışmak zorunda değil program eğer

programın girdisini destekleyen türde bir model bulur iseniz o modeli indirip programa attığınızda program modeli tanıyor ve model ile senkronize çalışabiliyor bu özellik daha geliştirme aşamasındadır bilindiği üzere her modelin farklı girdi boyutları farklı girdi metrikleri ve farklı farklı istekleri bağımlılıkları mevcuttur şu anlık sadece programın girdisine göre tahmin yapabilen modeller program tarafından destekleniyor ancak bunları kullanmak zorunda değilsiniz bu sadece bir alternatif zaten temelde sağlam iki tane model mevcut ancak modellerde hala geliştirme aşamasında;

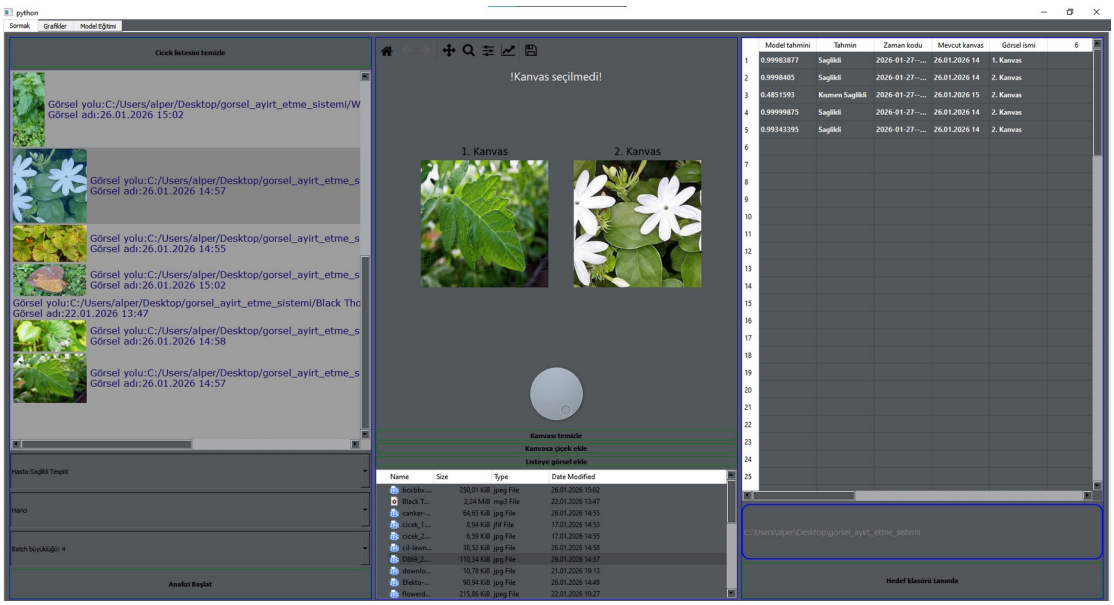
Ve şimdi ise programdaki asıl özelliğe değinelim; Programda ***Transfer Learning*** *mevcut yani programı kendiniz eğitebiliyorsunuz ve yaptığı hataları tekrarlamasını azaltabiliyorsunuz Şu anda bu özellik geliştirme aşamasında olsada programın bünyesi bunu kaldıracak mimariye sahip yani bu özellik eklenecek bir özellik ve teknik aksaklıklar çıkmaz ise programa kesin olarak eklenecek özelliklerden birisi bariz olarak budur; Progamda seçtiğiniz modelin* ***Summary*** *yani katman bilgilerini parametre sayılarını VB. bilgilerine erişmenizi sağlayan bir sistemde şu anda mevcut*

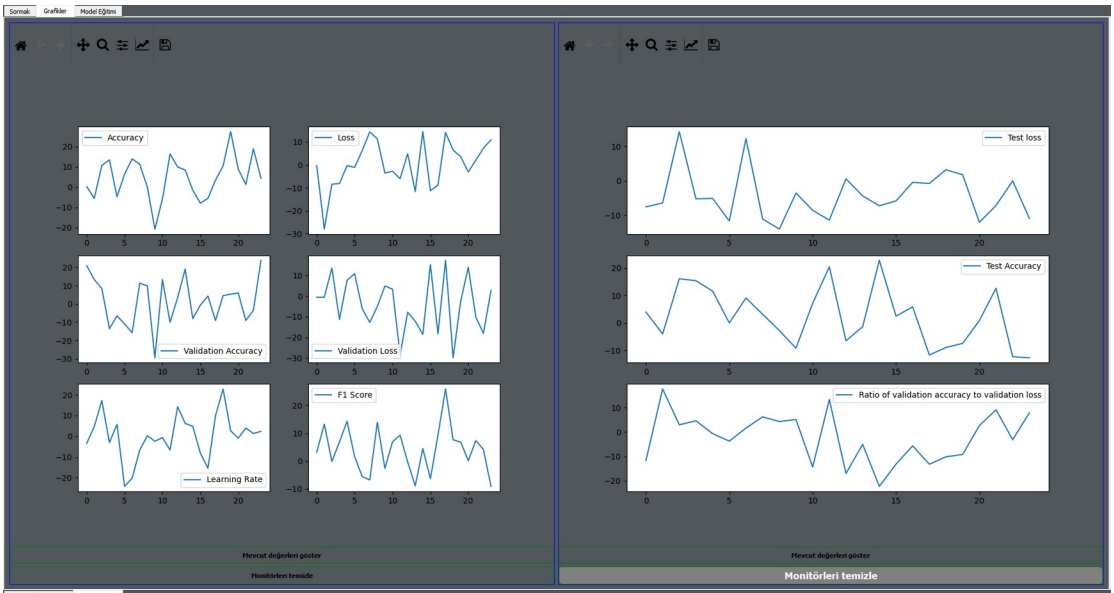
Ayrıca programdaki bir özellik ise model oluşturma sekmesi; evet bu sekme daha programda teknik olarak

mevcut değil ancak en kısa zamanda eklencek özelliklerden birisi ve teknik mimari bu özelliğe uygun tasarlandı

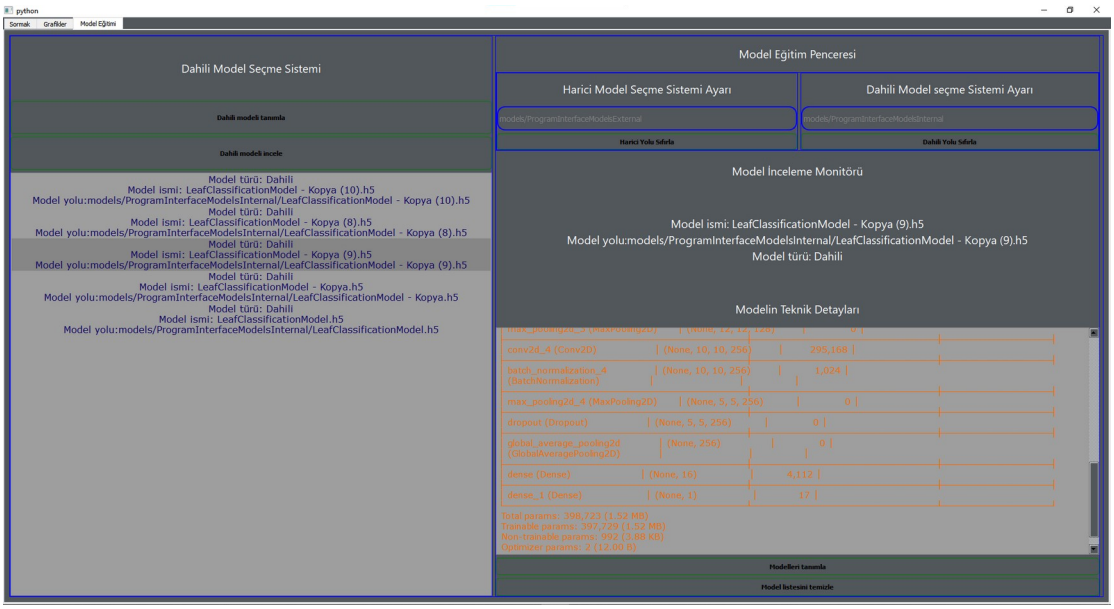
Ayrıca programdaki log sistemi sayesinde programı yazarken hazırladığım modeller dahil sizin eğiteceğiniz modellerinde tüm eğitim bilgilerinin loglandığı ve bu logların bir sistem aracılıgı ile parse edilip sayısal formata dönüştükten sonra grafiklerinin çıkarlıldığı bir sistemde programda Mevcuttur. Yani programda bir log + plotting sistemi mevcut olarak bulunmakta

Şimdi ise artık programın model ile olan bağlantıları ve özelliklerini anlatmayı bir kenara bırakıp programın kendisini sunayım;




Sormak
Grafikler
Model Eğitimi
Accuracy
Loss
Validation Accuracy
Validation Loss
F1 Score
Learning Rate
Mevcut değerleri göster
Monitörleri temizle
Test loss
Test Accuracy
Ratio of validation accuracy to validation loss
Mevcut değerleri göster
Monitörleri temizle

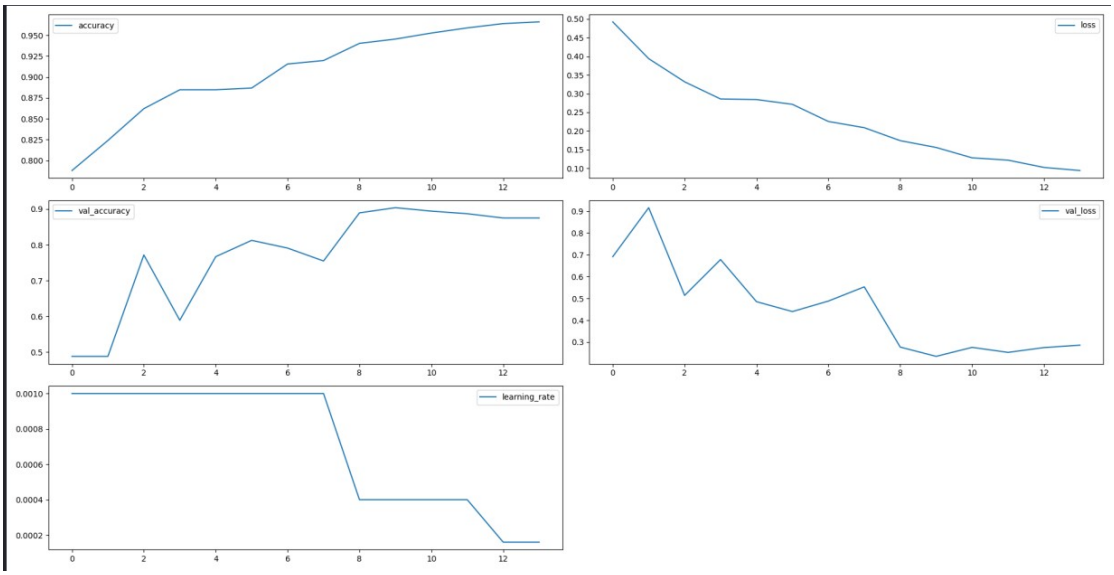

Sormak
Grafikler
Model Eğitimi
Dahili Model Seçme Sistemi
Dahili modeli tanımla
Dahili modeli incele
Model Eğitim Penceresi
Harici Model Seçme Sistemi Ayarı
Dahili Model seçme Sistemi Ayarı
models/ProgramInterfaceModelsExternal
models/ProgramInterfaceModelsInternal
Harici Yolu Sıfırla
Dahili Yolu Sıfırla
Model İnceleme Monitörü
Model ismi: LeafClassificationModel - Kopya (9).h5
Model yolu:models/ProgramInterfaceModelsInternal/LeafClassificationModel - Kopya (9).h5
Model türü: Dahili
Modelin Teknik Detayları
Model yolu:models/ProgramInterfaceModelsExternal/model.h5
Model ismi: model.h5
Model türü: Harici
Model: "sequential"
Layer (type)
Output Shape
Param #
conv2d (Conv2D)
(None, 222, 222, 16)
448
batch_normalization (BatchNormalization)
(None, 222, 222, 16)
64
max_pooling2d (MaxPooling2D)
(None, 111, 111, 16)
0
conv2d_1 (Conv2D)
(None, 109, 109, 32)
4,640
batch_normalization_1 (BatchNormalization)
(None, 109, 109, 32)
128
Modelleri tanımla
Model listesini temizle
Harici Model Seçme Sistemi
Harici modeli tanımla
Harici modeli incele



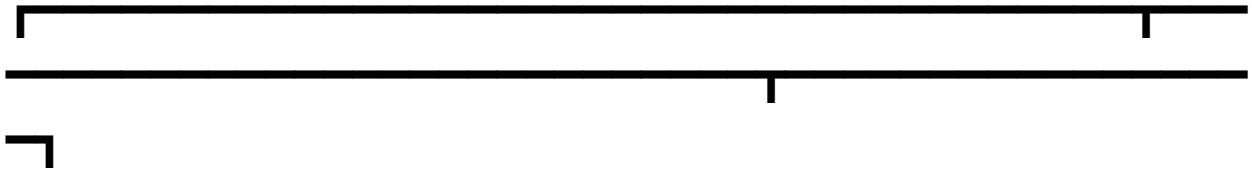
Görmüş olduğunuz üzere programda olduğunu belirttiğim özellikler bu şekilde gözüküyor ayrıca eklemeyi unuttuğum şey ise şu;

Modele tahmin yaptırmadan önce ***Batch büyüklüğü****,* ***Model türü****,* ***Model*** *gibi etkenleri seçip o yönde bir çıktı alabiliyorsunuz*

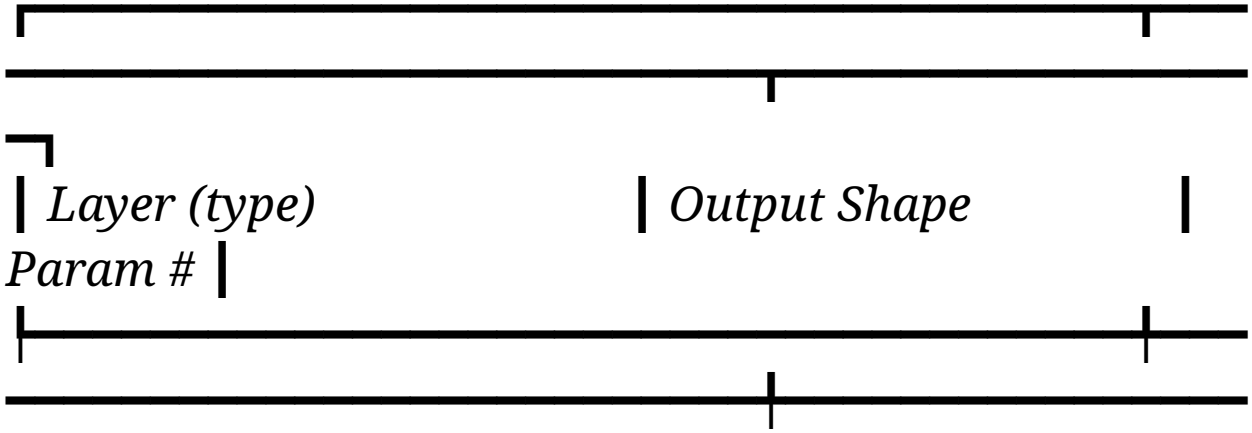
Şimdi ise modelin mevcut öğrenme grafiklerine gelelim



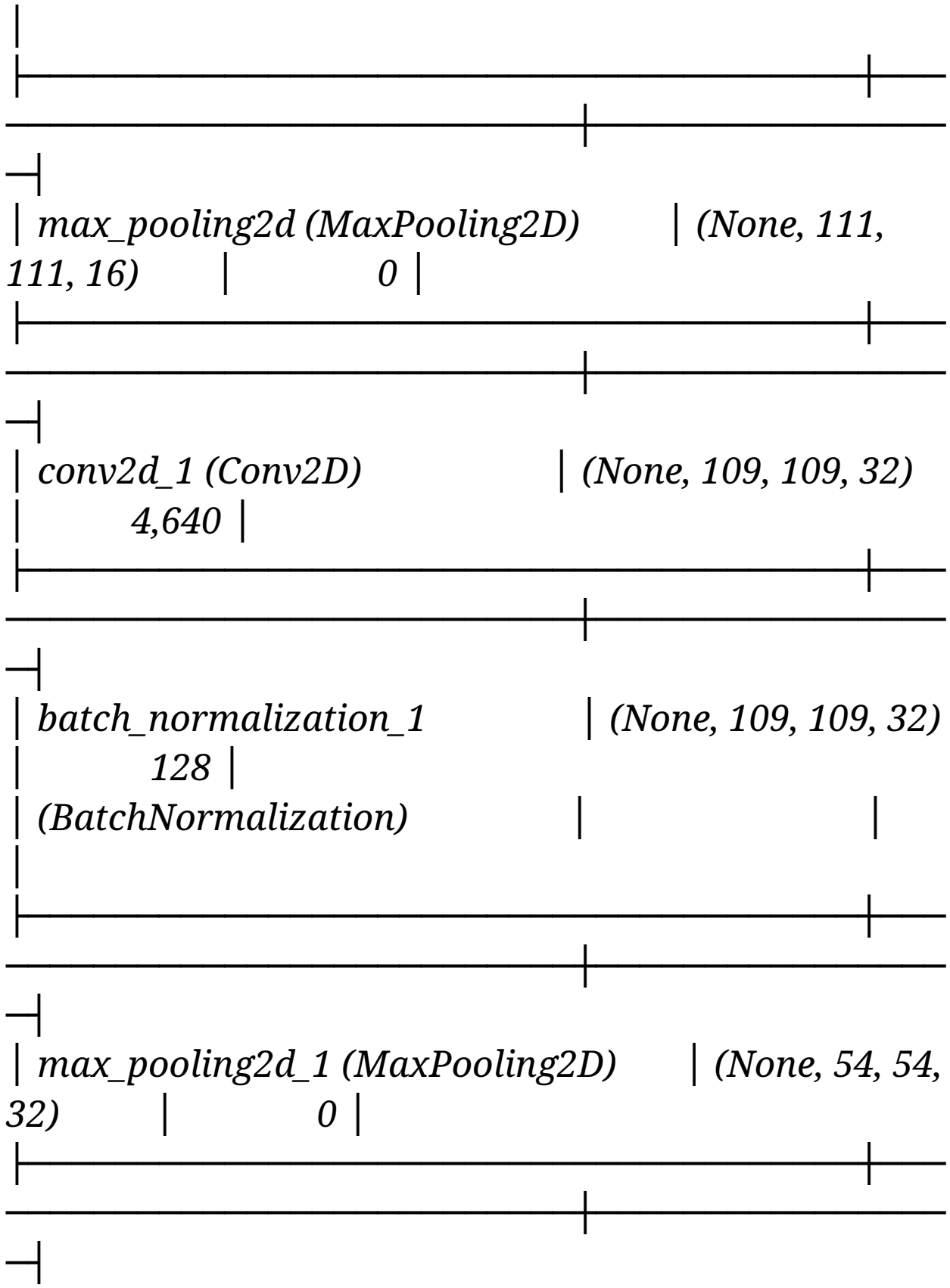
*Grafiklerdede görmüş olduğunuz üzere **accuracy ile loss** birbirine ters orantılı ancak göze çarpan bir etken var **Validation Accuracy** ve **Validation Loss** grafikleri bu grafikler inişli çıkışlı bir şekilde başlamış ancak ileri **epoks** değerlerinde kontrollü şekilde dengelenmiş buda veri setinin zor olduğunu ve modelin veri setinde zorlandığını gösteren etkenlerden biridir 8. **Epoks**ta **Learning Rate** değeri düşmüş ve o epokstada diğer **Validation Loss** artarken **Validation Accuracy** ise artış göstermiştir yani **ReduceLROPlateu** callback'i işini yapıyor demek ve 14 eposklu bir eğitim olmasına rağmen 14. epoksta durması **EarlyStopping** in çalıştığına işarettir modelin anlık durumu mükemmel değil ancak model şu anda yaprakların hasta olup olmadığını kaba hesap %75 doğrulukta tespit edebiliyor Şimdi ise modelin mimarisini aşşağıya bırakıyorum*

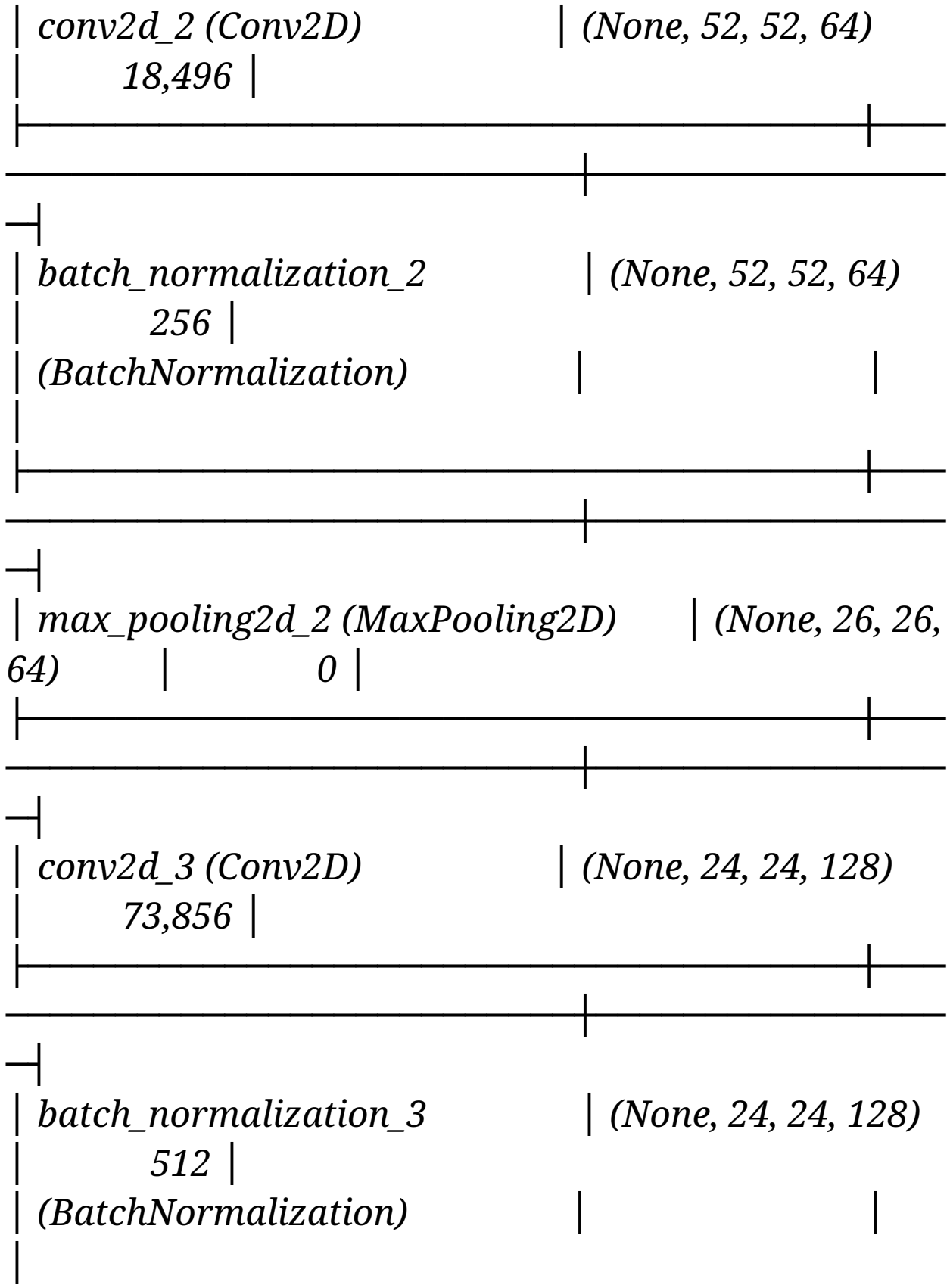


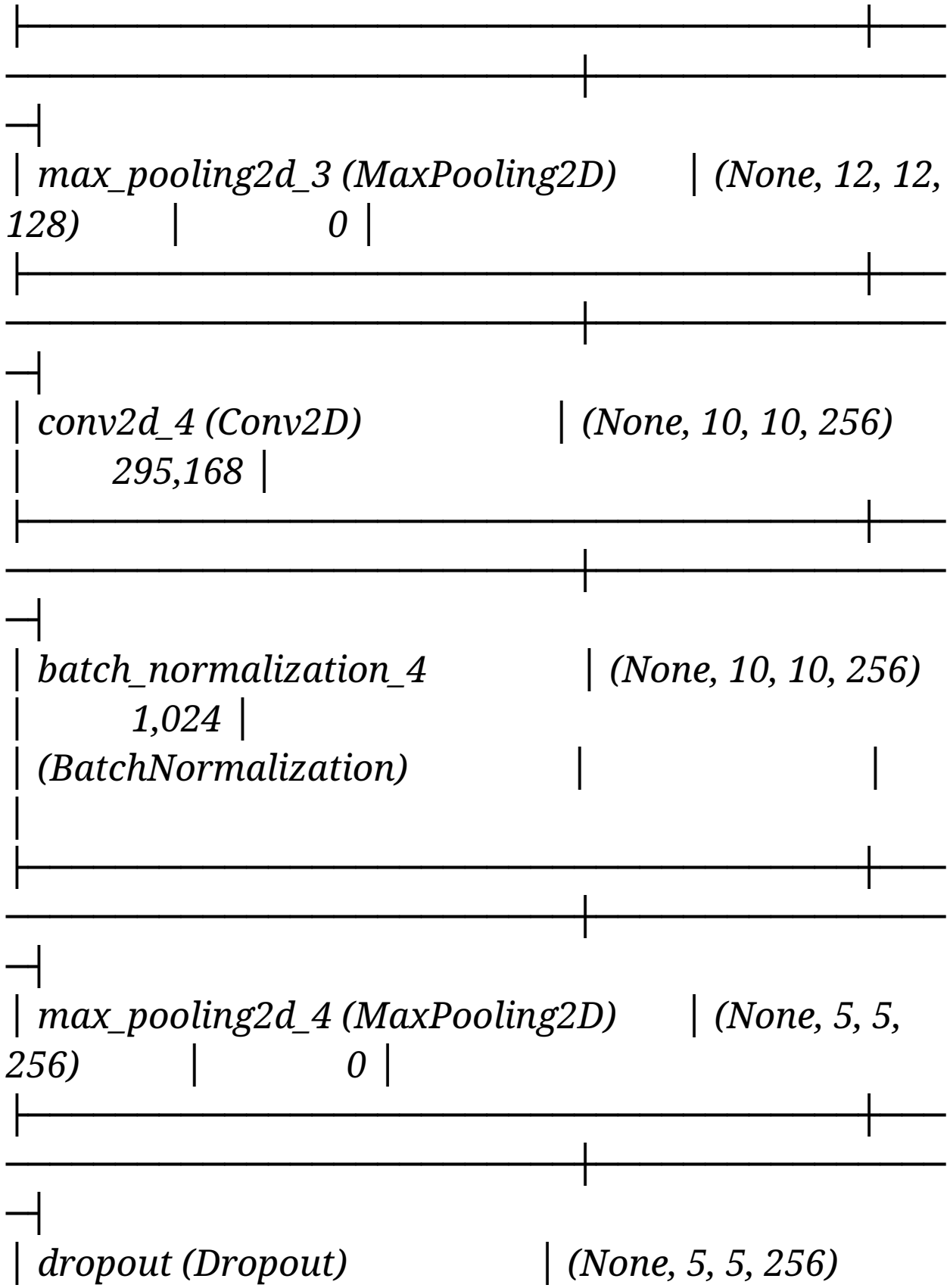
Model yolu:models/ProgramInterfaceModelsExternal/model.h5
Model ismi: model.h5
Model türü: Harici
Model: "sequential"

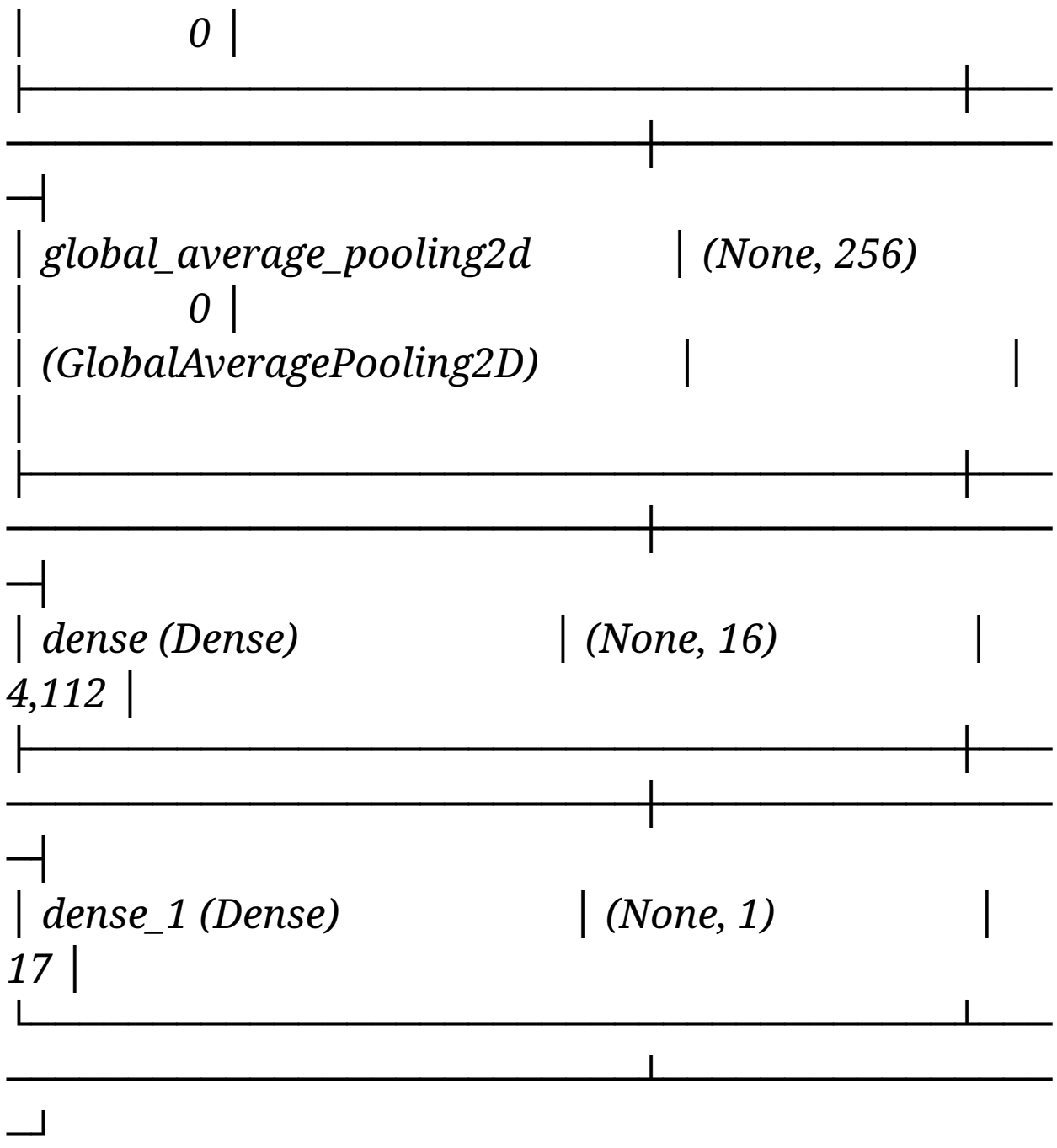


Layer (type)	*Output Shape*	*Param #*
conv2d (Conv2D)	*(None, 222, 222, 16)*	*448*
batch_normalization (BatchNormalization)	*(None, 222, 222, 16)*	*64*




conv2d_2 (Conv2D) (None, 52, 52, 64) 18,496
batch_normalization_2 (BatchNormalization) (None, 52, 52, 64) 256
max_pooling2d_2 (MaxPooling2D) (None, 26, 26, 64) 0
conv2d_3 (Conv2D) (None, 24, 24, 128) 73,856
batch_normalization_3 (BatchNormalization) (None, 24, 24, 128) 512





Total params: 398,723 (1.52 MB)
Trainable params: 397,729 (1.52 MB)
Non-trainable params: 992 (3.88 KB)
Optimizer params: 2 (12.00 B)