

KOCAELİ ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ

BİTİRME TEZİ

TÜMÖR SEGMENTASYONU

ÖĞRENCİ ADI SOYADI: RECEP KAYA
DANIŞMAN: MELTEM KURT PEHLİVANOĞLU

KOCAELİ 2023

İÇİNDEKİLER

İÇİNDEKİLER	2
ŞEKİLLER DİZİNİ	3
TABLolar DİZİNİ	4
GİRİŞ	5
1.LİTERATÜRDEKİ ÇALIŞMALAR	7
2.VERİ SETİ	8
3.GÖRÜNTÜ ÖNİŞLEME	10
4. MODELLER	11
4.1. Vgg19	13
4.2. Xception	15
4.3. Fcn	16
4.4. U-Net	18
4.5. Nasnet	21
5.Metrikler	23
5.1. Dice Skoru	23
5.2. İou Değeri	24
5.3. F1 Skoru	25
6.SONUÇLAR	26
KAYNAKÇA	28

ŞEKİLLER DİZİNİ

Şekil 4.1. Örnek VGG19 Mimarisi.....	13
Şekil 4.2. Örnek Xception Mimarisi.....	15
Şekil 4.3. Örnek FCN Mimarisi.....	17
Şekil 4.4. Örnek U-Net Mimarisi.....	19
Şekil 4.5. Örnek NASNET Mimarisi.....	21
Şekil 6.1. FCN Tahmin Görüntüsü.....	26
Şekil 6.2. U-Net Tahmin Görüntüsü.....	26
Şekil 6.3. VGG19 Tahmin Görüntüsü.....	27
Şekil 6.4. NASNET Tahmin Görüntüsü	27
Şekil 6.5. Xception Tahmin Görüntüsü	27

TABLolar DİZİNİ

Tablo 2.1. Eğitim , Doğrulama Ve Test Tablosu.....	8
Tablo 2.2. Veri Seti Örnek Görüntüleri.....	9
Tablo 6.1. Skorlar.....	26

GİRİŞ

Beyin tümörleri, dünya genelinde yaygın olarak görülen ve ciddi sağlık sorunlarına neden olan önemli bir hastalıktır. Tümörlerin erken teşhis edilmesi ve doğru bir şekilde segmente edilmesi, tedavi sürecinde önemli bir rol oynamaktadır. Geleneksel yöntemlerle beyin tümörlerinin segmentasyonu manuel olarak gerçekleştirilmekte ve uzmanlar tarafından yapılan görsel incelemelere dayanmaktadır. Ancak, bu yöntemler zaman alıcıdır, öznel ve hatalı sonuçlara yol açabilir.

Derin öğrenme modellerinin hızlı gelişimi, medikal görüntü analizinde yeni bir dönemi başlatmıştır. Derin öğrenme algoritmaları, karmaşık veri yapılarını anlamlandırma ve yüksek düzeyde otomatikleştirme yetenekleriyle dikkat çekmektedir. Bu nedenle, derin öğrenme modellerinin beyin tümörü segmentasyonunda kullanılması, daha doğru ve güvenilir sonuçlar elde etmek için umut verici bir yaklaşım olmuştur.

Bu tezde, derin öğrenme modellerinin beyin tümörünün segmentasyonunda kullanılması üzerine bir araştırma sunulmaktadır. Amaç, geleneksel yöntemlerin kısıtlamalarını aşarak, daha doğru ve otomatik bir segmentasyon yöntemi geliştirmektir. Tezin amacına ulaşmak için, farklı derin öğrenme mimarileri, özellik çıkarma yetenekleri kullanılarak deneysel bir çalışma gerçekleştirilecektir.

Bu tezde kullanılacak olan veri seti, beyin tümörü görüntülerini içeren birçok farklı modaliteye sahip olacaktır. Derin öğrenme modelleri, bu çok modaliteli görüntüler üzerinde eğitilecek ve beyin tümörlerini doğru bir şekilde segmente edebilmek için öğreneceklerdir. Ayrıca, geliştirilen modellerin performansları karşılaştırılarak değerlendirilecektir.

Bu tez, derin öğrenme modellerinin beyin tümörü segmentasyonunda kullanılabilirliğini ve etkinliğini değerlendirmeyi hedeflemektedir. Elde edilen sonuçlar, tıbbi görüntü analizinde derin öğrenme modellerinin potansiyelini ortaya koyarak, daha doğru ve hızlı bir teşhis ve tedavi sürecine katkıda bulunabilir. Ayrıca, geliştirilen segmentasyon yöntemi, tıbbi uzmanların teşhis ve tedavi planlama süreçlerinde daha güvenilir bir rehberlik sağlayabilir.

Bu tezde kullanılacak olan veri seti, büyük miktarda beyin tümörü görüntülerini içerecektir. Bu görüntülerin çeşitliliği, derin öğrenme modelinin genelleme yeteneğini test etmek için önemli olacaktır. Ayrıca, veri setindeki görüntülerin etiketlenmesi ve segmentasyon alanında uzmanlaşmış tıbbi uzmanlar tarafından doğrulanması sağlanacaktır.

Tezin yöntemi, öncelikle uygun bir derin öğrenme mimarisinin seçilmesini içerecektir. Farklı derin öğrenme mimarileri, performansları ve öğrenme yetenekleri açısından değerlendirilecektir. Ardından, seçilen mimari, eğitim aşamasında veri setine uyarlanacak ve tümör segmentasyonunu gerçekleştirecek şekilde eğitilecektir.

Eğitim aşamasının tamamlanmasının ardından, geliştirilen modelin performansı çeşitli değerlendirme metrikleri kullanılarak değerlendirilecektir. Dice skoru, iou değeri ve F1 skoru gibi metrikler, modelin segmentasyon başarısını objektif bir şekilde değerlendirmek için kullanılacaktır.

Sonuçlar, derin öğrenme modelinin beyin tümörü segmentasyonunda ne ölçüde etkili olduğunu ve geleneksel yöntemlere kıyasla avantajlarını ortaya koyacaktır. Geliştirilen modelin dice skoru, iou değeri ve f1 skorları ayrıca modellerin ürettikleri tümör tahmin görüntüleri verilecektir

Bu tez, derin öğrenme modellerinin beyin tümörü segmentasyonunda kullanılabilirliğini ve etkinliğini göstermeyi amaçlamaktadır.

1. LİTERATÜRDEKİ ÇALIŞMALAR

Süleyman UZUN ve Diğerleri (2022): Bu çalışmada MRI görüntüleri üzerinde U-Net mimarisi kullanılarak beyin tümör hücrelerinin tespit edilmesini sağlayan bir sistem gerçekleştirilmiştir. Çalışmada literatürde sıkça kullanılan global veri setlerinden BRATS veri seti kullanılmıştır. Yapılan çalışma sonucunda güvenilirliği kabul edilebilen %91,38'lik bir dice skoru elde edilmiştir[1].

Tarıkcın DOĞANAY ve Oktay YILDIZ (2022): Bu çalışmada, beyin MR görüntülerinden otomatik tümör tespiti için uçtan uca Çok Ölçekli Çok Düzeyli Ağ (Multi-Scale Multi-Level Network MM-Network) modeli önerilmiştir. Gerçekleştirilen çalışmada, UNet'teki evrişimli ağ seviyesinde çoklu uzamsal ölçeklerin küresel bağlamsal özelliklerini birleştirerek, ağlar boyunca özellik haritalarının boyutuna bağlı olarak alıcı alanın farklı oranlarda genişlemesini sağlayan genişletilmiş evrişim modülünden yararlanılmıştır. Yapılan deneysel çalışmalarda önerilen model ile yüksek doğrulukta tümör tespiti sağlanmıştır[2].

Isensee, F., Schell, M., & Pflueger, I. (2021): Bu çalışma, derin öğrenme modeli kullanarak manyetik rezonans görüntülerinden otomatik beyin tümörü segmentasyonunu hedeflemektedir. Yazarlar, 3D konvolüsyonel sinir ağlarını kullanarak bir segmentasyon modeli geliştirmişlerdir ve bu modelin performansını çeşitli veri setleri üzerinde değerlendirmişlerdir [3].

Zhou ve diğerleri (2019): Bu çalışma, U-Net tabanlı tam evrişimli sinir ağlarını kullanarak otomatik beyin tümörü tespiti ve segmentasyonunu ele alır. Araştırmacılar, U-Net'in genel mimarisini kullanarak, farklı görüntü modlarından oluşan bir veri setinde beyin tümörlerini tespit etmeyi ve segmente etmeyi amaçlarlar. Bu çalışma, U-Net'in beyin tümörü segmentasyonunda etkili bir şekilde kullanılabileceğini ve iyi performans sağladığını gösterir [4].

Sharma, G., Verma, R., & Bhateja, V. (2021): Brain tumor segmentation and survival prediction using deep learning - Bu çalışma, derin öğrenme yöntemlerini beyin tümörü segmentasyonu ve hayatta kalma tahminleri için kullanmayı amaçlamaktadır [5].

2. VERİ SETİ

Tez çalışmamda, beyin tümörü segmentasyonunu gerçekleştirmek için kullandığım veri seti, toplamda 3065 adet PNG formatında MR görüntüsü ve bu görüntülerin ait oldukları bölgelerin maskelerini içermektedir. Bu veri seti, beyin tümörü segmentasyonu için eğitim ve test amaçlı kullanılacaktır.

MR görüntüleri, baş bölgesinin MR görüntülemesi sırasında elde edilen görüntülerdir. Beyin tümörlerini tespit etmek ve sınıflandırmak için kullanılan bu görüntüler, beyin yapısının farklı bölgelerini temsil eder. Her bir MR görüntüsü, yüksek çözünürlüklü bir görüntüdür ve farklı yoğunluk değerlerine sahip piksellerden oluşur.

Maskeler, MR görüntülerinin her pikselinin beyin tümörüne ait olup olmadığını belirten etiketlenmiş görüntülerdir. Maskeler, görüntülerle aynı boyutta olup, her pikselin 0 veya 1 değerini alabilen bir etiket değeri içerir. 0 değeri, pikselin beyin tümörüne ait olmadığını, 1 değeri ise pikselin beyin tümörüne ait olduğunu gösterir. Maskeler, beyin tümörlerinin yerini ve boyutunu belirlemek için kullanılır ve modelin eğitim aşamasında gerçek etiketleri sağlar.

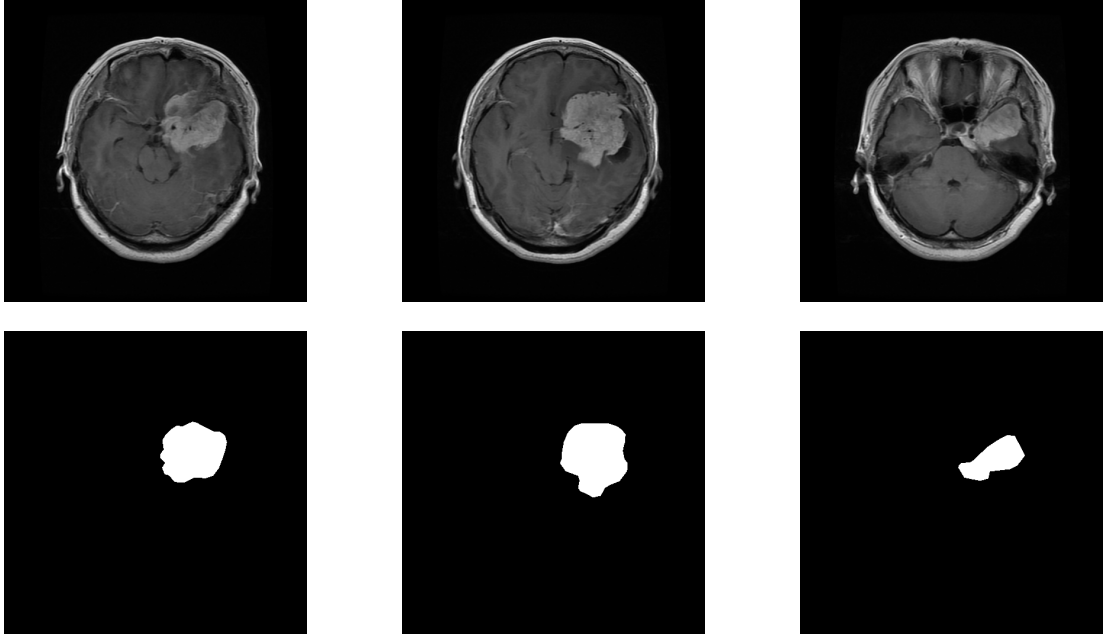
Veri seti model eğitiminde kullanabilmek için aşağıdaki gibi eğitim, doğrulama ve test parçalarına ayrılmıştır.

Tablo 2.1. Eğitim , Doğrulama Ve Test Tablosu

Eğitim	2482
Doğrulama	276
Test	307

Aşağıda veri setine ait örnek görüntüler verilmiştir:

Tablo 2.2. Veri Seti Örnek Görüntüleri



3.GÖRÜNTÜ ÖNİŞLEME

Dizindeki görüntü dosyaları yüklenir. Her bir görüntü, gri tonlamalı olarak okunur ve boyutu (256, 256) olarak yeniden şekillendirilir. Ardından, görüntü 3 kanallı (RGB) hale getirilir ve bir liste içine eklenir. Son olarak liste numpy dizisine dönüştürülür.

Bu bölüm, belirtilen dizindeki maske dosyalarını yükler. Her bir maske, gri tonlamalı olarak okunur ve boyutu (256, 256) olarak yeniden şekillendirilir. Ardından, maske değerleri 0-255 aralığından 0-1 aralığına ölçeklendirilir ve bir liste içine eklenir. Son olarak liste numpy dizisine dönüştürülür.

LabelEncoder sınıfı kullanılarak maske değerleri kategorik etiketlere dönüştürülür.Maskelerin şekli yeniden şekillendirilerek piksel değerleri tek boyutlu bir diziye dönüştürülür. Etiket kodlaması yapılır ve maske değerleri kategorik etiketlere dönüştürülür. Etiketler, orijinal şekillerine yeniden şekillendirilir.

4. MODELLER

Modellerde aşağıdaki parametreler kullanılmıştır:

LR = 0.0001: Bu parametre, öğrenme oranını temsil eder. Öğrenme oranı, ağırlıkların ne kadar hızlı veya yavaş öğrenildiğini kontrol eder. Daha yüksek bir öğrenme oranı daha hızlı öğrenmeyi sağlarken, düşük bir öğrenme oranı daha istikrarlı ancak daha yavaş bir öğrenme sağlar. LR = 0.0001, düşük bir öğrenme oranı kullanmayı tercih ettiğinizi gösterir. Bu, modelin daha istikrarlı bir şekilde öğrenmesini sağlar.

optimizer = Adam(LR): Bu parametre, optimize ediciyi belirtir. Adam, gradient tabanlı bir optimize edici algoritmadır. Gradient tabanlı optimize ediciler, ağırlıkların güncellemek için gradyanları kullanır. Adam optimize edici, adaptif momentum yöntemini kullanarak gradyanların momentumunu hesaplar ve öğrenme oranını adaptif olarak ayarlar. Adam optimize edici, genellikle hızlı ve etkili bir şekilde öğrenme sağlar. LR = 0.0001 ile birlikte Adam optimize ediciyi kullanmayı tercih ettiğiniz için modelin daha iyi performans göstermesi ve daha hızlı bir şekilde öğrenmesi beklenir.

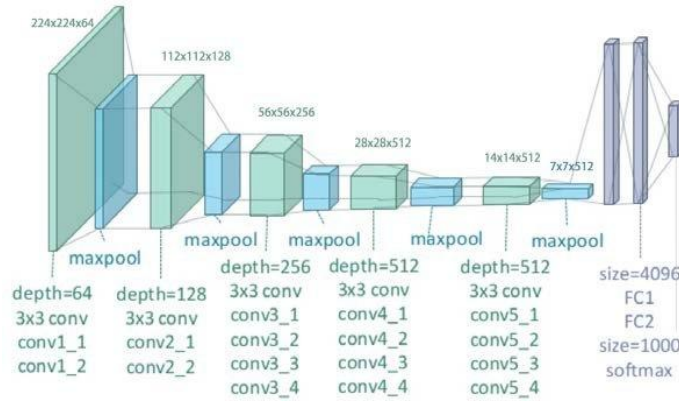
activation = 'softmax': Bu parametre, aktivasyon fonksiyonunu belirtir. Softmax aktivasyon fonksiyonu, çok sınıflı sınıflandırma problemleri için yaygın olarak kullanılan bir fonksiyondur. Softmax fonksiyonu, her sınıfın olasılığını hesaplar ve sınıflar arasında bir olasılık dağılımı oluşturur. Bu, modelin her bir sınıf için olasılıklı tahminler yapmasını sağlar.

loss = 'binary_crossentropy': Bu parametre, kayıp fonksiyonunu belirtir. Binary crossentropy, ikili sınıflandırma problemleri için kullanılan bir kayıp fonksiyonudur. Modelin tahminleri ile gerçek etiketler arasındaki farkı hesaplar ve bu farkı minimize etmeye çalışır. Binary crossentropy, iki sınıflı sınıflandırma problemleri için uygun bir kayıp fonksiyonudur.

Bu parametreler haricinde modeller SGD optimizer parametresi ve sigmoid aktivasyon parametresiyle çalıştırıldı ama tahmin görüntüsü alınmadığı için sadece yukarıdaki parametrelerin çıktıları sonuçlar kısmında verildi.

4.1. VGG19

VGG19, görüntü sınıflandırma görevleri için tasarlanmış bir evrişimli sinir ağı modelidir. Evrişimli katmanlar ve pooling katmanları kullanılarak daha derin ve karmaşık özelliklerin öğrenilmesi sağlanır. Tam bağlantılı katmanlar ise özelliklerin yüksek seviyeli temsilini oluşturarak sınıflandırma kararını verir. Modeldeki evrişimli katmanlar farklı filtrelerle giriş görüntüsünün özelliklerini algılar ve pooling katmanları boyut azaltma işlemi yaparak özelliklerin genelleştirilebilir hale gelmesini sağlar. Son katmanda ise softmax aktivasyon fonksiyonu kullanır[7].



Şekil 4.1 VGG19 Örnek Mimarisi

VGG19 modelinin önceden eğitilmiş ağırlıklarının, ImageNet veri kümesinde eğitilmiş ağırlıklardan yüklenmiştir. Model aşağıdaki yöntemler kullanılarak özelleştirilmiştir.

x = UpSampling2D(size=(4, 4))(base_model.output): Bu satır, base_model'in çıktısını 4 kat büyüten bir up-sampling işlemi uygular. Bu, özellik haritalarının boyutunu artırarak daha yüksek çözünürlüklü bir segmentasyon haritası elde etmeyi amaçlar.

`x = Conv2D(256, 3, activation='relu', padding='same')(x)`: Bu satır, 256 filtreli ve 3x3 boyutunda bir evrişimli katman uygular. Bu katman, özellik haritalarını daha derinleştirerek daha kapsamlı bilgi yakalamayı hedefler.

`x = UpSampling2D(size=(4, 4))(x)`: Bu satır, önceki çıktıyı tekrar 4 kat büyüten bir up-sampling işlemi uygular. Bu, özellik haritalarının boyutunu daha da artırarak segmentasyon için daha detaylı bir sonuç elde etmeyi amaçlar.

`x = Conv2D(128, 3, activation='relu', padding='same')(x)`: Bu satır, 128 filtreli ve 3x3 boyutunda bir evrişimli katman uygular. Bu katman, özellik haritalarını daha da derinleştirir ve segmentasyon performansını artırmayı amaçlar.

`x = UpSampling2D(size=(2, 2))(x)`: Bu satır, önceki çıktıyı 2 kat büyüten bir up-sampling işlemi uygular. Bu, özellik haritalarının boyutunu artırır ve daha ayrıntılı segmentasyon sonuçları elde etmeyi hedefler.

`x = Conv2D(64, 3, activation='relu', padding='same')(x)`: Bu satır, 64 filtreli ve 3x3 boyutunda bir evrişimli katman uygular. Bu katman, özellik haritalarının daha da derinleştirilmesini ve segmentasyon performansının iyileştirilmesini sağlar.

`outputs = Conv2D(num_classes, 1, activation='softmax')(x)`: Bu satır, num_classes sayısı kadar filtreli ve 1x1 boyutunda bir evrişimli katman uygular. Bu katman, her piksel için sınıflandırma yaparak segmentasyon sonuçlarını oluşturur. Softmax aktivasyon fonksiyonu kullanılarak piksel bazlı bir olasılık dağılımı elde edilir.

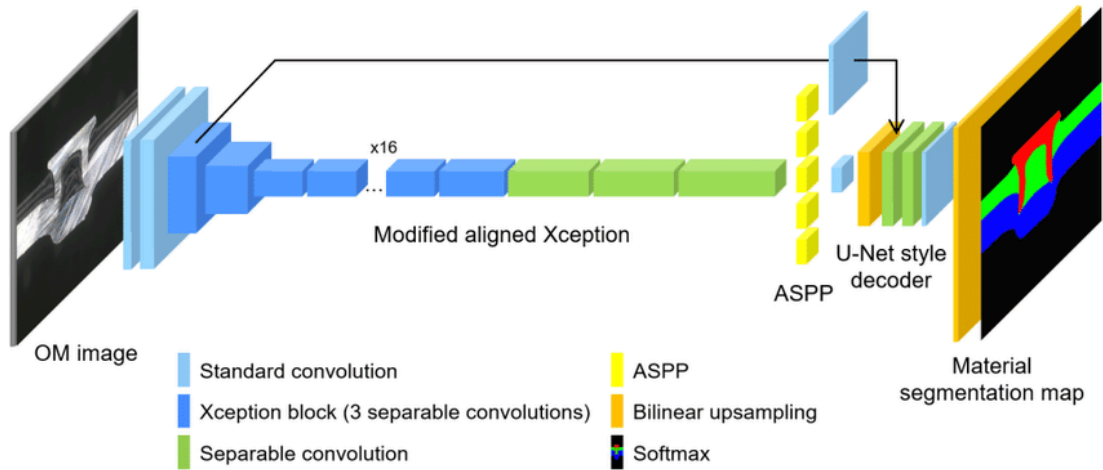
4.2. Xception

Xception (Extreme Inception) modeli, evrişimli sinir ağı (CNN) mimarilerinden biridir.

Xception'ın temel fikri, evrişim işlemini daha derin bir seviyede gerçekleştirmek ve böylece daha fazla özellik çıkarımı yapmaktır. Bu, bilgi akışını daha iyi kontrol etmeyi ve daha ayrıntılı özellikleri daha iyi yakalamayı sağlar.

Modeldeki ana bileşen, "derinlik ayrışması" olarak adlandırılan yapıdır. Bu yapıda, evrişim işlemi, kanal bazında ve piksel bazında iki ayrı aşamada gerçekleştirilir. Kanal bazında evrişim, 1x1 boyutunda evrişim filtreleri kullanılarak gerçekleştirilir. Ardından, piksel bazında evrişim, standart 3x3 boyutunda evrişim filtreleri kullanılarak gerçekleştirilir[10].

Bu model de imagenet veri seti ağırlıklarının VGG19 modelinde olduğu gibi özelleştirilerek kullanılmıştır.



Şekil 4.2. Xception Örnek Mimarisi

4.3.FCN

FCN (Fully Convolutional Network), piksel tabanlı görsel işleme görevlerinde kullanılan bir evrişimli sinir ağı modelidir. Geleneksel CNN'lerden farklı olarak, FCN evrişimli katmanlardan oluşur ve her pikseli tahmin etmek için kullanılır. Evrişim ve pooling işlemleriyle özellik haritaları oluşturulur ve yeniden örnekleme işlemiyle giriş boyutuna getirilir. Modelde evrişim ve pooling katmanlarından oluşan bir kodlama kısmı bulunur ve özellik haritaları yeniden örnekleme ile boyutlandırılır. Son olarak, her piksel için sınıf etiketi tahmini yapılır [11].

FCN modeli aşağıdaki gibi kullanıldı:

Modelin girişine input_shape parametresi ile belirtilen boyutta bir görüntü verilir. Bu örnekte, giriş görüntüler 256x256 piksel boyutunda ve 3 kanala sahip RGB görüntüleridir.

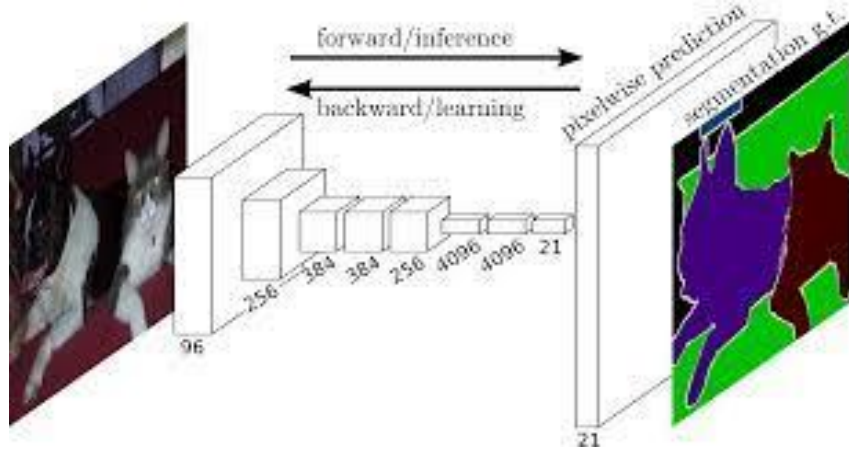
Modelin ilk katmanı, girişe uygulanan bir 2D evrişim (Conv2D) katmanıdır. Bu katmanda 64 adet filtre kullanılır ve ReLU aktivasyon fonksiyonu kullanılarak çıktı üretilir. Ardından, ikinci bir 2D evrişim katmanı aynı filtre sayısı ve aktivasyon fonksiyonu ile eklenir. Bu adımlar, görüntülerin özelliklerini çıkarmak ve temsil etmek için tekrarlanır.

Ardından, örnekleme (MaxPooling2D) katmanları eklenir. Bu katmanlar, özellik haritalarının boyutunu küçültmek ve önemli bilgileri korumak için kullanılır. Örneğin, ilk örnekleme katmanı 2x2 boyutunda bir pencere ile örnekleme yaparak özellik haritalarının boyutunu yarıya indirir.

Daha sonra, özellik haritalarının boyutunu artırmak için 2D evrişim katmanları ve örnekleme katmanları arasında tekrarlanan bir yapı kurulur. Bu yapı, özellik haritalarının daha geniş bir bağlamı kapsamalarını sağlar.

Modelin çıkış kısmında, son evrişim katmanının çıktıları üzerine bir 2D evrişim (Conv2D) katmanı uygulanır. Bu katmanda, 2 sınıflı temsil etmek için 2 filtre kullanılır ve softmax aktivasyon fonksiyonu kullanılarak piksel tabanlı bir olasılık dağılımı elde edilir.

Son olarak, oluşturulan model, Model sınıfı kullanılarak giriş ve çıkışları belirtilerek oluşturulur ve model değişkenine atanır.



Şekil 4.3. FCN Örnek Mimarisi

4.4 U-Net

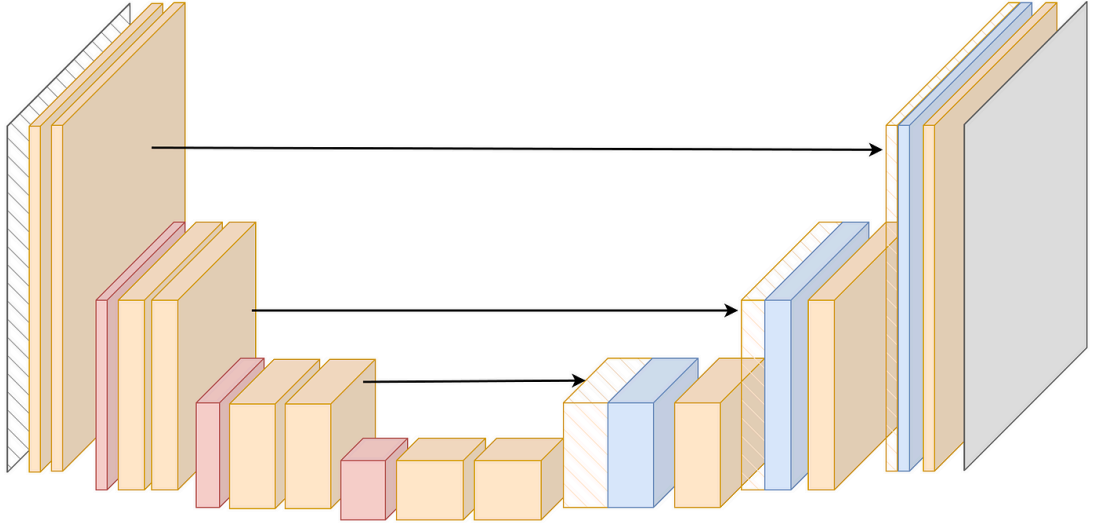
U-Net, görüntü segmentasyonu gibi piksel tabanlı görsel işleme görevlerinde kullanılan bir evrişimli sinir ağı (CNN) mimarisidir.

U-Net'in temel fikri, hem lokalize edici hem de bağlamsal bilgileri birleştiren ve yüksek çözünürlüklü tahminler üreten bir yapı oluşturmaktır.

U-Net, birçok evrişimli ve pooling katmanından oluşan bir encoding kısmına sahiptir. Bu kısım, giriş görüntüsünün düşük seviyedeki özelliklerini kodlar ve özellik haritalarını çıkış boyutuna doğru indirger. Evrişim ve pooling işlemleri, özelliklerin özetlenmesini ve önemli bilgilerin çıkarılmasını sağlar.

Modelin kodlama kısmından sonra decoding kısmı gelir. Bu kısım, özellik haritalarını artırarak giriş boyutuna getirir. Her artırma aşamasında, kodlama kısmından elde edilen özellik haritaları ile artırılan özellik haritaları birleştirilir. Bu sayede, hem düşük hem de yüksek seviyedeki özelliklerin birleştirilmesi sağlanır.

U-Net'in decoding kısmı, artırılan özellik haritalarının evrişimli katmanlardan geçirilmesi ve ardından özelliklerin daha ayrıntılı çıkarılması için evrişim işlemlerinin uygulanması ile devam eder. Bu işlemler, her artırma aşamasında daha yüksek çözünürlüklü tahminler üretmek için yapılan bir özelliktir.



Şekil 4.4. U-Net Örnek Mimarisi

Son katmanda, genellikle 1x1 boyutunda evrişimli katmanlar kullanarak sınıflandırma yapılır. Bu, her pikselin ilgili sınıfa ait olma olasılığını tahmin etmek için kullanılır[9].

Bu çalışmada U-Net modeli aşağıdaki gibi kullanılmıştır:

1. Giriş katmanı oluşturulur.
2. İlk downsample bloğu oluşturulur: İlk konvolüsyon bloğu çifti uygulanır. Ardından, özellik haritalarını boyutu yarıya indiren bir MaxPooling işlemi uygulanır.
3. İkinci downsample bloğu oluşturulur: İkinci konvolüsyon bloğu çifti uygulanır. Özellik haritalarını boyutu yarıya indiren bir MaxPooling işlemi uygulanır.

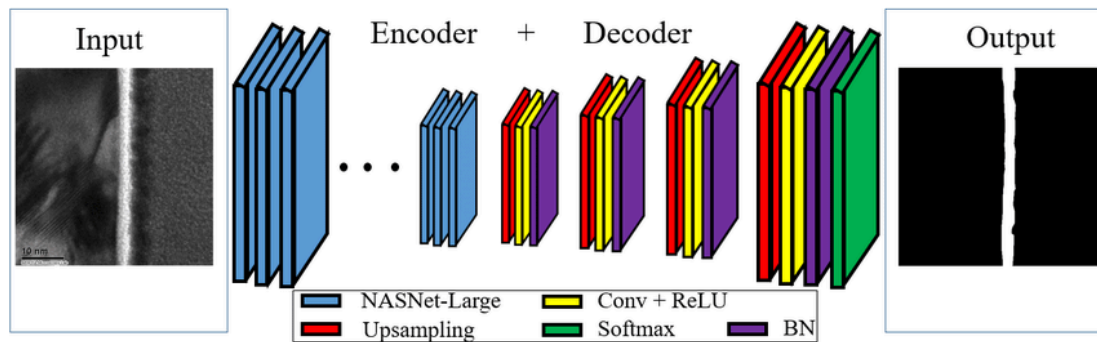
4. Üçüncü downsample bloğu oluşturulur: Üçüncü konvolüsyon bloğu çifti uygulanır. Özellik haritalarını boyutu yarıya indiren bir MaxPooling işlemi uygulanır.
5. Dördüncü downsample bloğu oluşturulur: Dördüncü konvolüsyon bloğu çifti uygulanır. Özellik haritalarını boyutu yarıya indiren bir MaxPooling işlemi uygulanır.
6. Bottleneck (darboğaz) katmanı oluşturulur: İki ardışık konvolüsyon bloğu uygulanır.
7. İlk upsample bloğu oluşturulur: Transpoz konvolüsyon işlemi ile boyutu 2 katına çıkarılır. Özellik haritası ile birleştirilir ve ardından iki ardışık konvolüsyon bloğu uygulanır.
8. İkinci upsample bloğu oluşturulur (7. adımdan benzer şekilde).
9. Üçüncü upsample bloğu oluşturulur (8. adımdan benzer şekilde).
10. Dördüncü upsample bloğu oluşturulur (9. adımdan benzer şekilde).
11. Son olarak, çıktı katmanı oluşturulur: Dördüncü upsample bloğunun çıktısı ile 1x1 boyutunda bir filtre uygulanır. Bu işlem, çıktıyı belirtilen sınıf sayısına uygun hale getirir. Oluşturulan model, giriş ve çıkış katmanlarıyla birlikte tanımlanır ve döndürülür.

4.5 NASNET

Mikro ve Makro Döğümler: NASNet, bir hücre içinde bir dizi mikro döğüm ve makro döğüm kullanır. Mikro döğümler, hücre içindeki bağlantı ve katman düzenini belirlerken, makro döğümler, hücrelerin birbiriyle nasıl bağlandığını ve nasıl tekrarlandığını belirler. Bu döğümler, modelin mimarisini tanımlamak için kullanılan temel yapı birimleridir.

NASNet, bir dizi özellik çıkarıcı bloğu içerir. Bu bloklar, veriye özgü özellikleri çıkarmak için giriş verileri üzerinde bir dizi konvolüsyon, aktivasyon fonksiyonu ve havuzlama işlemi uygular. Özellik çıkarıcı blokları, daha yüksek seviyeli özellikleri yakalamak için girdi verilerini daha karmaşık bir şekilde işler.

Modelin sonunda, çıkarılan özelliklerin temsili bir vektöre dönüştürülmesi için global ortalama havuzlama kullanılır. Bu işlem, tüm kanalların özellik haritasındaki piksel değerlerinin ortalamasını alarak yapılan bir havuzlama işlemidir. Bu, boyutu sabit bir vektör elde etmek için kullanılır[12].



Şekil 4.5. Örnek NASNET Mimarisi

Global ortalama havuzlamadan sonra, tam bağlantılı (fully connected) katmanlar yer alır. Bu katmanlar, çıkarılan özelliklerle ilişkilendirilmiş sınıfları tahmin etmek için kullanılır. Sınıflandırma problemleri için tipik olarak softmax aktivasyon fonksiyonuyla sonuç üretilir.

NASNet, özellik çıkarıcı blokları içeren bir yapay sinir ağıdır. Bu bloklar, veriye özgü özellikleri çıkarmak için konvolüsyon, aktivasyon fonksiyonu ve havuzlama işlemlerini kullanır. Mikro düğümler ve makro düğümler, hücre içindeki bağlantı ve katman düzenini belirler. Özellik çıkarıcı bloklarının çıktıları, global ortalama havuzlama ile bir vektöre dönüştürülür. Bu vektör, verinin özünü temsil eder. Son olarak, tam bağlantılı katmanlar, çıkarılan özellikleri sınıflandırmak için kullanılır ve tahminler yapılır. NASNet, bu yapıyla veriden özellikleri çıkarır, temsil edici bir vektör elde eder ve sınıflandırma tahmini yapar.

Bu model de imagenet veri seti ağırlıklarının VGG19 modelindeki gibi özelleştirilerek kullanılmıştır

5.METRİKLER

5.1. DICE Skoru

Dice skoru (Dice score), genellikle görüntü segmentasyonu veya nesne tanıma gibi alanlarda kullanılan bir değerlendirme metriğidir. Bu skor, modelin tahminlerinin gerçek etiketlerle ne kadar iyi eşleştiğini ölçer ve 0 ile 1 arasında değer alır.

Dice skoru, tahmin edilen ve gerçek etiketler arasındaki örtüşme alanının, bu iki alanın toplam alanına oranını temsil eder. Özellikle dengesiz sınıf dağılımına sahip veri kümesi üzerinde çalışırken, doğru pozitif ve yanlış pozitif tahminleri dikkate alarak bir modelin performansını değerlendirmek için yaygın olarak kullanılır.[6]

Dice skoru hesaplama formülü aşağıdaki gibidir:

$$\text{Dice} = (2 * TP) / (2 * TP + FP + FN)$$

Burada:

TP (True Positive): Gerçek pozitiflerin sayısı (doğru tahmin edilen piksel sayısı)

FP (False Positive): Yanlış pozitiflerin sayısı (yanlış tahmin edilen piksel sayısı)

FN (False Negative): Yanlış negatiflerin sayısı (tahmin edilmeyen gerçek piksel sayısı)

5.2. IoU Değeri

IOU (Intersection over Union), görüntü segmentasyonu veya nesne tanıma gibi alanlarda kullanılan bir değerlendirme metriğidir. IOU, tahmin edilen bölgenin gerçek bölgeyle ne kadar iyi örtüştüğünü ölçer. Aynı zamanda Jaccard Index olarak da bilinir[8].

IOU değeri, tahmin edilen bölge ile gerçek bölge arasındaki ortak alanın, bu iki alanın toplam alanına oranını ifade eder. Bu metrik 0 ile 1 arasında değer alır, 1'e yaklaştıkça daha iyi bir uyumu temsil eder.

IOU değerini hesaplamak için aşağıdaki formül kullanılır:

$$\text{IOU} = (\text{Intersection area}) / (\text{Union area})$$

Burada:

Intersection area: Tahmin edilen bölge ile gerçek bölgenin kesişim alanı.

Union area: Tahmin edilen bölge ile gerçek bölgenin birleşim alanı.

5.3. F1 Skoru

F1 skoru, sınıflandırma modellerinin performansını değerlendirmek için kullanılan bir metriktir. Precision (kesinlik) ve recall (hatırlama) metriklerini birleştirilerek hesaplanır. F1 skoru, bir modelin hem kesinlikle hem de hatırlamayla ne kadar iyi çalıştığını gösterir.

Precision, doğru pozitif tahminlerin toplam pozitif tahminlere oranını ifade eder. Precision, yanlış pozitifleri minimize etmeye odaklanır ve yanlış pozitif tahminlerin etkisini azaltır. Bir modelin kesinlik değeri ne kadar yüksekse, doğru pozitif tahminlerin oranı o kadar yüksek olur.

Recall, gerçek pozitif tahminlerin toplam gerçek pozitiflere oranını ifade eder. Recall, yanlış negatifleri minimize etmeye odaklanır ve yanlış negatif tahminlerin etkisini azaltır. Bir modelin hatırlama değeri ne kadar yüksekse, gerçek pozitif tahminlerin oranı o kadar yüksek olur.

F1 skoru, precision ve recall değerlerinin harmonik ortalamasını temsil eder. Harmonik ortalamayı kullanmanın nedeni, precision ve recall değerleri arasındaki dengenin korunmasını sağlamaktır. F1 skoru, hem precision hem de recall değerlerinin yüksek olması durumunda en yüksek değeri alır. Bu nedenle, bir modelin hem precision hem de recall değerlerini optimize ettiği durumlarda F1 skoru maksimum olur.

F1 skoru aşağıdaki formülle hesaplanır:

$$\text{precision} = \text{TP} / (\text{TP} + \text{FP})$$

$$\text{recall} = \text{TP} / (\text{TP} + \text{FN})$$

$$\text{F1} = 2 * (\text{precision} * \text{recall}) / (\text{precision} + \text{recall})$$

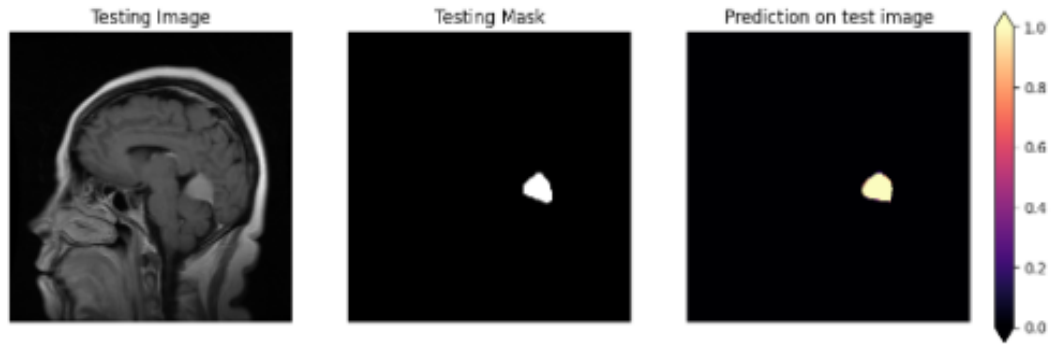
6. SONUÇLAR

Tablo 6.1. Test Sonuçları

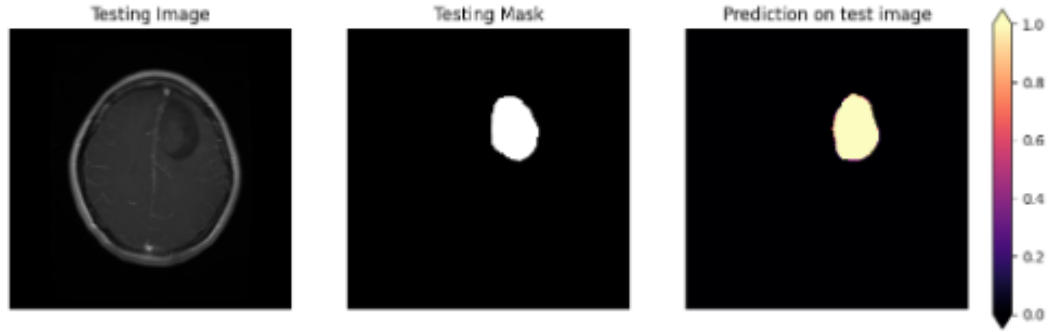
Model Adı	Dice Skoru	IoU değeri	F1 Skoru
FCN	91.23	98.96	99.48
Unet	90.02	98.80	99.41
VGG19	90.20	98.80	99.42
Xception	88.84	98.65	99.34
NASNetMobile	88.21	98.56	99.30

Tablo 6.1. Doğrulama Verisi Sonuçları

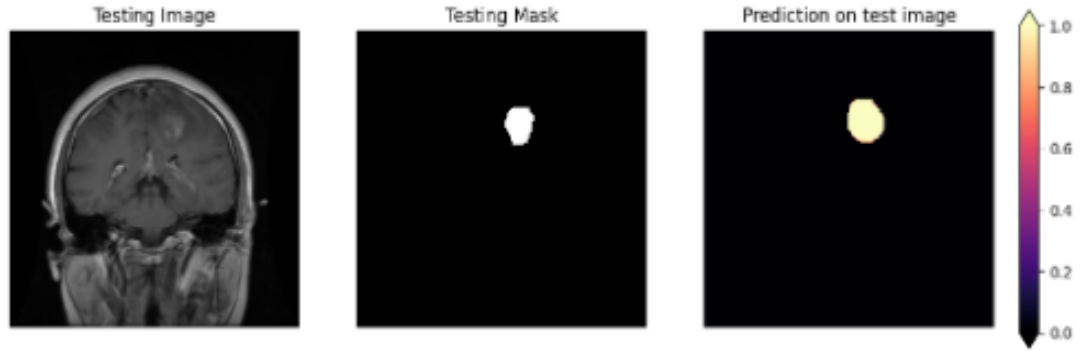
Model Adı	Dice Skoru	IoU değeri	F1 Skoru
FCN	91.10	99.90	99.45
Unet	89.91	98.76	99.39
VGG19	89.04	98.61	99.34
Xception	88.83	98.62	99.33
NASNetMobile	87.84	98.49	99.26



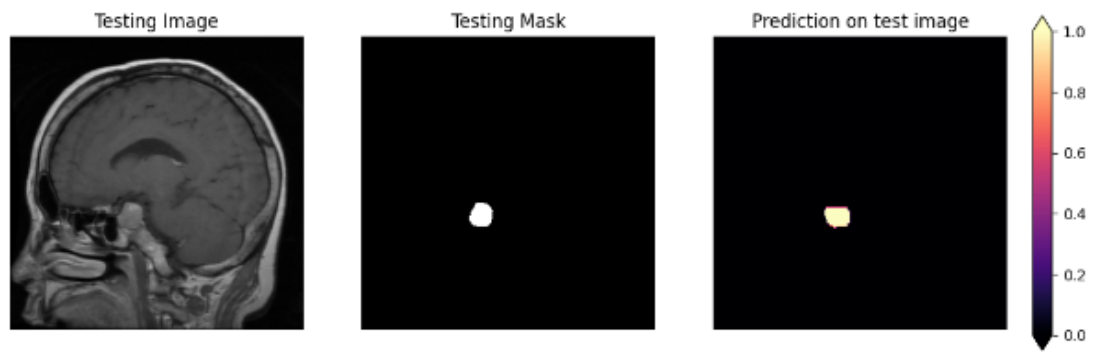
Şekil 6.1. FCN Tahmin Görüntüsü



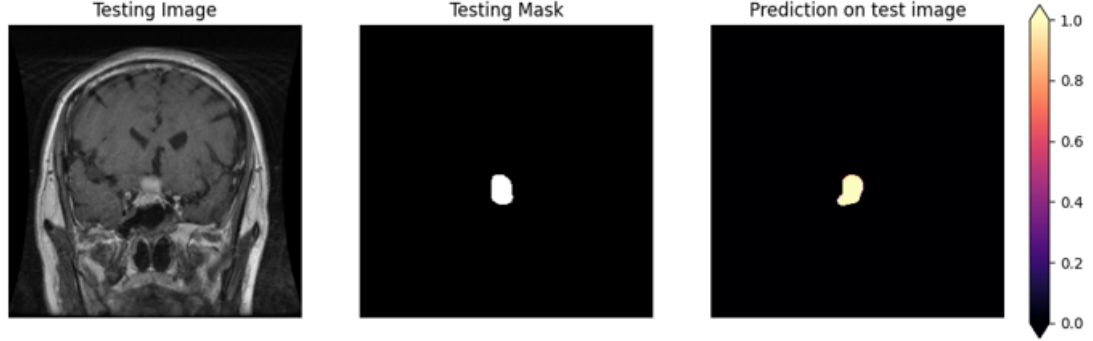
Şekil 6.2. U-Net Tahmin Görüntüsü



Şekil 6.3. VGG19 Tahmin Görüntüsü



Şekil 6.4. NASNET Tahmin Görüntüsü



Şekil 6.5. Xception Tahmin Görüntüsü

KAYNAKÇA

- [1] S. Uzun , E. Güney and B. Bingöl , "U-Net Mimarisi ile Beyin Tümörü MRI Görüntülerinin Segmentasyonu", *El-Cezeri*, vol. 9, no. 4, pp. 1583-1590, Dec. 2022, doi:10.31202/ecjse.1169424
- [2] T. Doğanay and O. Yıldız , "Beyin Tümör Tespiti İçin Derin Öğrenme Temelli Bilgisayar Destekli Tanı Sistemi", *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, vol. 10, no. 4, pp. 1748-1762, Oct. 2022
- [3] Isensee, F., Schell, M., & Pflueger, I. (2021). Automated brain tumor segmentation based on MRI using deep learning. in 2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI), 82-85. doi: 10.1109/ISBI48211.2021.9434001
- [4] Zhou, Y., Chang, H., Yang, W., & Yu, J. (2019). Automatic brain tumor detection and segmentation using U-Net based fully convolutional networks. arXiv preprint arXiv:1901.09959.
- [5] Sharma, G., Verma, R., & Bhateja, V. (2021). Brain tumor segmentation and survival prediction using deep learning. *Journal of Ambient Intelligence and Humanized Computing*, 12(5), 5479-5494. doi: 10.1007/s12652-020-02623-2

- [6] “Dice score - OECD.AI.” *OECD AI Policy Observatory*, <https://oecd.ai/en/catalogue/metrics/dice-score>. Accessed 29 May 2023.
- [7] “AlexNet, VGGNet, Inception ve ResNet Nedir? | by Kaan Bıçakcı | Medium.” *Kaan Bıçakcı*, 17 December 2020, <https://frightera.medium.com/alexnet-vggnet-inception-ve-resnet-nedir-bddc7482918b>. Accessed 29 May 2023.
- [8] “Intersection over Union (IoU).” *Hasty.ai*, <https://hasty.ai/docs/mp-wiki/metrics/iou-intersection-over-union>. Accessed 29 May 2023.
- [9] “U-Net.” *Wikipedia*, <https://en.wikipedia.org/wiki/U-Net>. Accessed 29 May 2023.
- [10] *Xception — With Depthwise Separable Convolution, Better Than Inception-v3 (Image....* (2018, September 25). Towards Data Science. Retrieved May 29, 2023, from <https://towardsdatascience.com/review-xception-with-depthwise-separable-convolution-better-than-inception-v3-image-dc967dd42568>
- [11] “FCN — Fully Convolutional Network (Semantic Segmentation) | by Sik-Ho Tsang.” *Towards Data Science*, <https://towardsdatascience.com/review-fcn-semantic-segmentation-eb8c9b50d2d1>. Accessed 29 May 2023.
- [12] “RNASNet — Neural Architecture Search Network (Image Classification).” *Sik-Ho Tsang*, <https://sh-tsang.medium.com/review-nasnet-neural-architecture-search-network-image-classification-23139ea0425d>. Accessed 29 May 2023.