

**T.C.**  
**SAKARYA ÜNİVERSİTESİ**  
**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**

**BSM 498 BİTİRME ÇALIŞMASI**

**GÖRÜNTÜ İŞLEME İLE TASARIMLARI HTML  
KODLARINA ÇEVİRME**

**G140910054 - Ertan ÇAKIR**

**Fakülte Anabilim**

**Dalı** : **BİLGİSAYAR MÜHENDİSLİĞİ**

**Tez Danışmanı** : **Doç. Dr. Cüneyt BAYILMIŞ**

**2018-2019 Bahar Dönemi**



T.C.  
SAKARYA ÜNİVERSİTESİ  
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

**GÖRÜNTÜ İŞLEME İLE TASARIMLARI HTML  
KODLARINA ÇEVİRME**

**BSM 498 - BİTİRME ÇALIŞMASI**

**Ertan ÇAKIR**

**Fakülte Anabilim Dalı : BİLGİSAYAR MÜHENDİSLİĞİ**

**Bu tez .. / .. / ... tarihinde aşağıdaki jüri tarafından oybirliği / oyçokluğu ile kabul edilmiştir.**

.....  
**Jüri Başkanı**

.....  
**Üye**

.....  
**Üye**

## ÖNSÖZ

Günümüzde yapay zekâ teknolojisi hızla gelişmektedir. Ülkemizin de bu alanlarda desteklerini arttırması, üniversitelerimizin çalışmalarını bu alanlara yönlendirmesi son derece umut vericidir. Yapay zekâ ve makine öğrenmesi kullanarak yapılmış olan bu çalışmada tecrübelerini benimle paylaşan danışman hocam Doç. Dr. Cüneyt BAYILMIŞ'a teşekkürlerimi sunarım.

## İÇİNDEKİLER

ÖNSÖZ	iv
İÇİNDEKİLER	v
SİMGELER VE KISALTMALAR LİSTESİ	v
ŞEKİLLER LİSTESİ	vii
TABLolar LİSTESİ	viii
ÖZET	ix
BÖLÜM 1. GİRİŞ	1
1.1. Proje Amacı	1
1.2. Daha Önce Yapılmış Çalışmalar	1
1.3. Projede Kullanılacak HTML Etiketleri	2
BÖLÜM 2. YAZILIM	3
2.1. Görüntü İşleme	3
2.2. Derin Öğrenme	3
2.1. İlgili Kütüphaneler ve Geliştirme Ortamı	4
2.2. Sürüm Denetim Sistemi	5
BÖLÜM 3. PROJE İŞLEYİŞİ VE TASARIMI	6
3.1. Veri Seti ve Derin Öğrenme Modeli	7
3.2. Görüntünün İşlenmesi	9
3.3. Tanınan Etiketlerin HTML Koduna Çevrilmesi	10
BÖLÜM 4. SONUÇLAR VE ÖNERİLER	11
KAYNAKLAR	12
ÖZGEÇMİŞ	13

## **SİMGELER VE KISALTMALAR LİSTESİ**

RGB	: Red Green Blue
YSA	: Yapay Sinir Ağları
Html	: Hyper Text Markup Language

## ŞEKİLLER LİSTESİ

Şekil 1 Bir görüntünün farklı şekilde işlenmiş halleri	3
Şekil 2 Perceptron Modeli	4
Şekil 3 Sistem İşleyiş Diyagramı	6
Şekil 4 Usecase Diyagram	7
Şekil 5 Verilen bir görüntü üzerindeki tanınan şekiller	7
Şekil 6 Derin öğrenme modeli katmanları	8
Şekil 7 Eğitimdeki doğruluk artış ve kayıp düşüş grafikleri	8
Şekil 8 Görüntünün işlenme akış diyagramı	9

## **TABLÖLAR LİSTESİ**

Tablo 1 Etiket Tablosu

2



## ÖZET

Anahtar kelimeler: Görüntü İşleme, Html, Mockup, Tasarım, Arayüz

Web sitesi yapma konusunda en çok zorlanılan kısımlardan birisi tasarımıdır. Tasarımın kullanıcıya hitap etmesi açısından en ince ayrıntısına kadar özenle yapılması gerekmektedir. Bu proje ile tasarım yapma konusunda yazılımcıya kolaylık sağlamayı ve kullanıcı isteklerini daha hızlı ve rahat karşılamayı amaçlandı. Proje mockup yazılımlarının kullandığı standartlaşmış modülleri kullanmaktadır.

Proje tahta ya da düz bir zemin üzerine çizilen mockup tasarımlarını HTML kodlarına çevirmektedir. Zemin üzerinden kamera ile çekilen görüntü, işlenmekte ardından bilgisayar ortamına aktarılmaktadır. Bilgisayara aktarılan görüntünün işlendikten sonra içerisindeki mockup tasarımları belirli bir formata uyarlanarak kayıt altında tutulmaktadır. Bu mockup tasarımları daha sonra belirlenen bir format içerisindeki her bir mockup tasarımının koordinatlarına uygun bir şekilde kod parçaları olarak HTML tasarımı çıktısı vermektedir. Bu çıktı kullanılan arayüz yardımıyla daha sonra değiştirilerek tasarımın kontrolü sağlanmaktadır

## **BÖLÜM 1. GİRİŞ**

Bir uygulamayı tasarlarken yapılması gerekenlerin başında arayüz tasarımı gelmektedir. Çünkü bir projenin arayüzü projenin ilgili müşterileri ya da kullanıcılarının o uygulamayı kullanmaya devam etmesi için çok önemlidir. Bu sebeple bir uygulamanın kodlarını yazmaya başlamadan önce tasarımlarının bir çizim programında ya da el ile kâğıt üzerinde çizilip sonrasında bu tasarımların arayüz kodları yazılır ve bu işlemler fazladan süre kaybına neder olur. Bu yüzden bu işlemlerin kolaylaştırılıp tasarımların doğrudan HTML kodlarına çevrilmesi bu süre kaybını ortadan kaldırabilir.

### **1.1. Proje Amacı**

Projenin çalışma şekli html mockup tasarımlarının html kodlarına çevrilmesidir. Tasarımları gerçek zamanlı olarak algılamak için görüntü işleme kullanıldı. Proje tasarımı python diline göre tasarlandı. Esasen proje iki kısımdan oluşmakta birinci kısım görüntüyü işlemek ve belirli bir formata çevirmek. İkinci kısım ise bu formatı kullanarak html kodlarını oluşturmak ve bunları ekranda gerçek zamanlı olarak göstermek



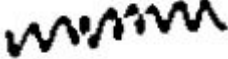


### **1.2. Daha Önce Yapılmış Çalışmalar**

Proje konusuna benzer olarak daha önceden github üzerinde bir çalışma bulunmakta bu çalışmada projeyi tasarlayan kişi internet sitelerinin görüntülerini rastgele oluşturup bunlardan veri seti elde etmekte sonrasında bu veriler ile derin öğrenme ağını eğitip kendi yazdığı arayüz üzerine bir görüntü yükleyerek bu görüntüden html, ios ve android için çıktı elde etmekte. Proje için bir dökümantasyon bulunmadığı için detaylı incelenemedi.

### 1.3. Projede Kullanılacak HTML Etiketleri

Projede sistemin tanınması gereken çizimler ve html etiket karşılıkları aşağıdaki tabloda verilmiştir.

Tablo 1 Etiket Tablosu

Html Kodu	Etiket Adı	Tasarım Çizimi
<code>&lt;img src = "image.png"&gt;</code>	Resim Etiketi	
<code>&lt;p&gt;Lorem Ipsum&lt;/p&gt;</code>	Paragraf Etiketi	
<code>&lt;h1&gt;Başlık&lt;/h1&gt;</code>	Başlık Etiketi	
<code>&lt;button type="button"&gt;Click&lt;/button&gt;</code>	Buton Etiketi	
<code>&lt;input type = "text"&gt;</code>	TextBox Etiketi	

Header etiketi, Paragraf etiketi ile şekil olarak aynıdır. Boyut olarak farklıdır. (Yükseklik 50 birimden fazla ise)

## BÖLÜM 2. YAZILIM

### 2.1. Görüntü İşleme

Görüntü işleme, bir görüntünün ihtiyaca uygun olarak matematiksel işlemler ile başka bir görüntüye çevrilmesine denir. Her bir görüntü RGB kodlarından meydana gelir bu RGB kodlarındaki değişiklikler görüntülerin farklılaşmasına yol açar. Görüntü işlemede tam olarak bu şekilde yapılır.

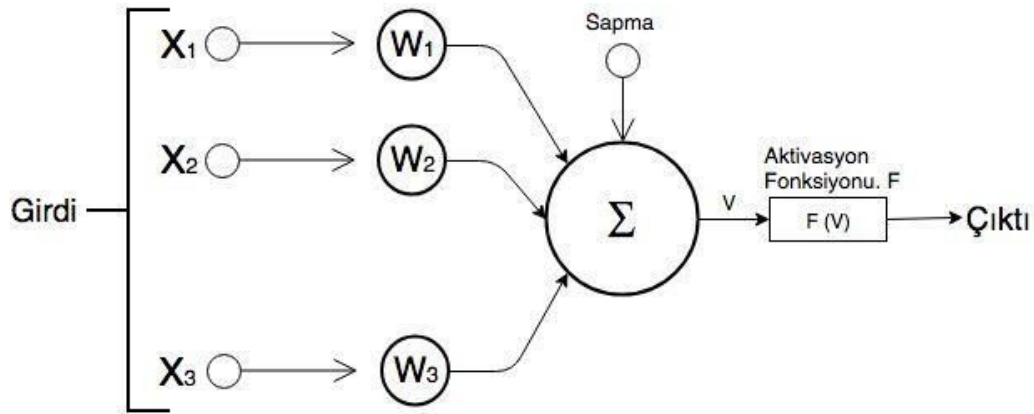


Şekil 1 Bir görüntünün farklı şekilde işlenmiş halleri

Görüntü işlemenin başlıca kullanılma sebebi orijinal görüntü üzerinden bilgi edinmek zordur bu yüzden görüntü üzerindeki gerekli bilgileri almadan önce gereksiz bilgiler görüntüden çıkarılmalıdır.

### 2.2. Derin Öğrenme

Derin öğrenme ise aslında yapı olarak yapay sinir ağlarına benzemektedir. Yapay sinir ağları ise temel olarak insan beynini bilgisayar üzerinde matematiksel olarak modellenmesidir. YSA modelinin içerisinde insan beyninde olduğu gibi nöronlar bulunmaktadır.



Şekil 2 Perceptron Modeli

Şekil 3 de görüldüğü gibi verilen girdiler ağırlık değerleri ile çarpılıp toplanarak aktivasyon fonksiyonuna verilir. Aktivasyon fonksiyonuna göre bir çıktı üretilir. Sonrasında eğer çıktı istenilen çıktıdan farklı ise nöronlar üzerindeki ağırlıklar güncellenir.

Avantajlarına rağmen YSA'nın en büyük dezavantajı öğrenme süresinin uzun sürmesi ve katman sayısının az olmasıdır. Bu yüzden Derin Öğrenme modeli ortaya atılmıştır. Derin öğrenme YSA'dan farklı olarak daha fazla katmana ve nörona sahip olabilir ve matematiksel hesaplamalar işlemciler yerine ekran kartları üzerinde yapılır. Ekran kartlarında işlemcilerden daha fazla çekirdek olduğu için öğrenme süreleri YSA'na göre daha kısadır.

## 2.1. İlgili Kütüphaneler ve Geliştirme Ortamı

Proje linux ortamında ve python dili ile geliştirilmiştir. Python dilinde yazılan bir programın diğer işletim sistemlerinde de çalışabiliyor olması sebebi ile windows ortamında da proje çalışabiliyor. Python dilinin seçilme sebebi derin öğrenme ve makine öğrenmesi alanında çok kullanılması ve bu sebeple hem kütüphanelerin hem de kaynakların daha çok olmasıdır.

Kullanım kolaylığı ve proje geliştirilirken çıkan sorunlar için internette bulunan kaynak fazlalığından derin öğrenme modelinin tasarımı için tercih edilen kütüphane keras oldu. Derin öğrenme modellerinin birkaç satır kod ile tasarlanabiliyor olması da kütüphane seçiminde etkili olmuştur.

Kütüphane alt yapı olarak tensorflow altyapısı kullanıyor bu sayede hesaplamalar kullanılan bilgisayar üzerindeki ekran kartı üzerinde hesaplama yapma imkanı sağlıyor.

Görüntü işleme için OpenCV kütüphanesi kullanıldı. Yüklenen görüntünün eşik değeri (threshold) değerlerinin alınması bu kütüphane ile sağlanmıştır. Yine aynı kütüphane yardımı ile yüklenen görüntü üzerinde kenarlar yardımıyla nesnelerin tespiti sağlanmıştır.

Ayrıca başta bir python kütüphanesi olan numpy kullanılmıştır. Görüntüler üzerinde oynama, verilerin saklaması ve taşınması bu kütüphane yardımı ile sağlanmıştır.

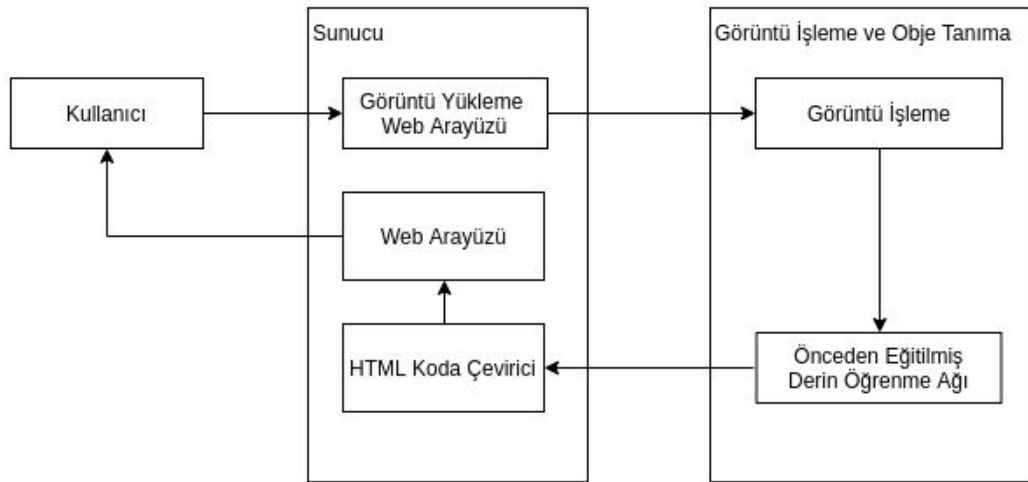
## **2.2. Sürüm Denetim Sistemi**

Proje geliştirilirken versiyon kontrol sistemi olarak önce gitlab kullanılmıştır. Daha sonra proje github'a taşınmıştır.

Github linki : <https://github.com/ertancakir/mockup-recognizer>

### BÖLÜM 3. PROJE İŞLEYİŞİ VE TASARIMI

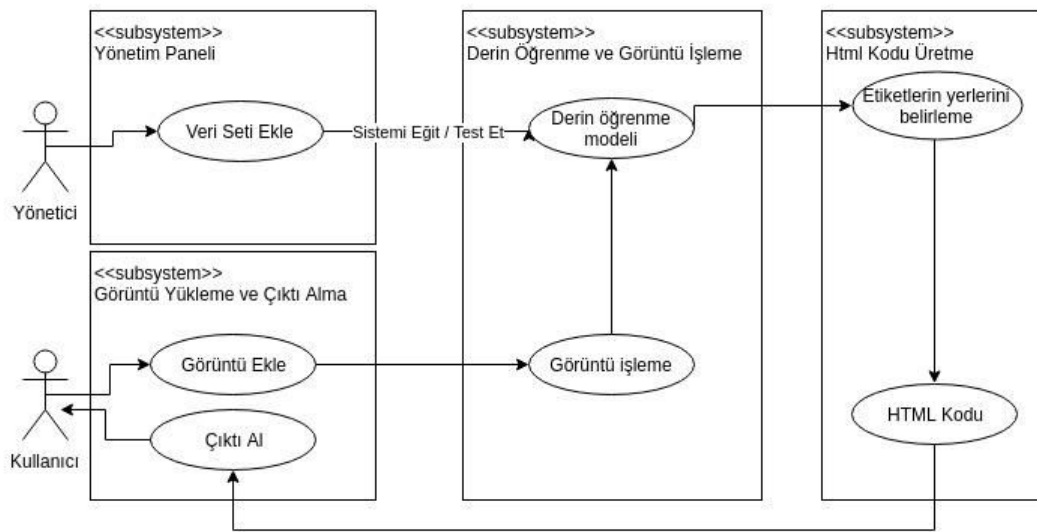
Sistem kullanıcıdan bir görüntü istenmektedir. Kullanıcı görüntüyü web arayüzü üzerinden sisteme yükler. Sunucu üzerinde görüntü hsv modeline çevrilir. Siyah ve beyaz görüntü yoğunluklarına göre görüntü parçalanır ve her bir parça daha önce eğitilmiş olan derin öğrenme modeline verilir. Sonrasında modelden verilen çıktı ile görüntü üzerindeki çizimlerin koordinatları hesaplanarak birleştirilir ve json formatı halinde html koda çevirmek için başka bir sisteme verilir. Burada json formatı işlenir ve html kod çıktısı verilir.



Şekil 3 Sistem İşleyiş Diyagramı

Web arayüzü üzerinden kullanıcının çizdiği görüntü ve html çıktısı aynı anda verilir.

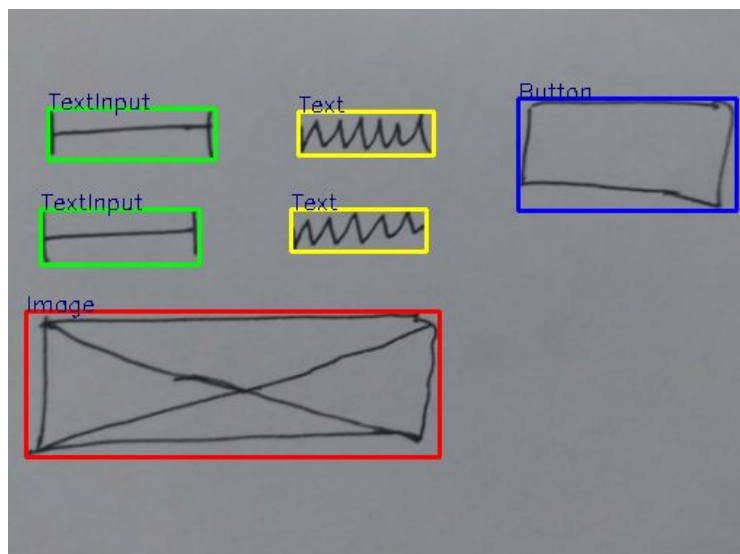
Bu aşamada kullanıcı .html uzantısı olarak çıktıyı indirebilir.



Şekil 4 Usecase Diyagram

### 3.1. Veri Seti ve Derin Öğrenme Modeli

Derin öğrenme yapay sinir ağlarından farklı olarak daha fazla nöron ve katmandan oluşmaktadır bu yüzden modelleri eğitmek için büyük miktarda veriye ihtiyaç duyulmaktadır. Fakat mevcut projede az bir veri seti ile iyi sonuç elde edildiği için veri setini büyükmek gerekmemiştir.



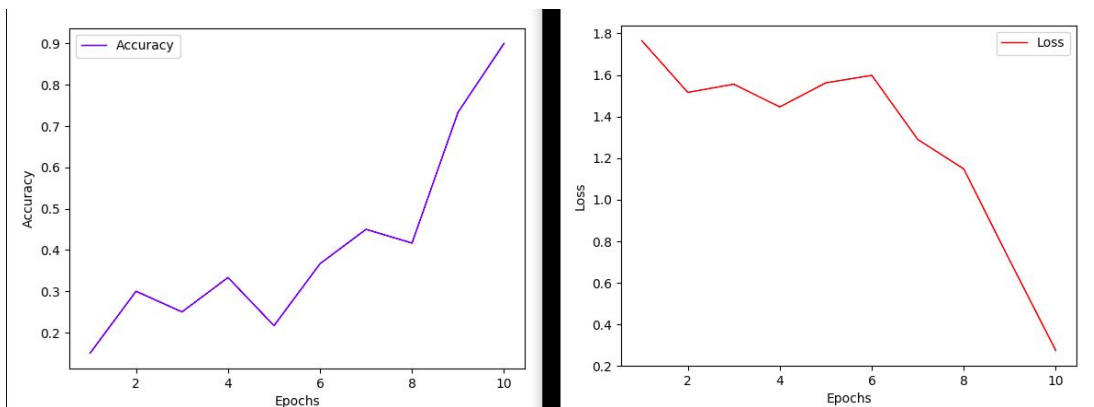
Şekil 5 Verilen bir görüntü üzerindeki tanınan şekiller



Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 80, 120, 16)	448
conv2d_2 (Conv2D)	(None, 78, 118, 16)	2320
max_pooling2d_1 (MaxPooling2D)	(None, 39, 59, 16)	0
dropout_1 (Dropout)	(None, 39, 59, 16)	0
conv2d_3 (Conv2D)	(None, 39, 59, 32)	4640
conv2d_4 (Conv2D)	(None, 37, 57, 32)	9248
max_pooling2d_2 (MaxPooling2D)	(None, 18, 28, 32)	0
dropout_2 (Dropout)	(None, 18, 28, 32)	0
flatten_1 (Flatten)	(None, 16128)	0
dense_1 (Dense)	(None, 512)	8258048
dropout_3 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 4)	2052
Total params: 8,276,756		
Trainable params: 8,276,756		
Non-trainable params: 0		

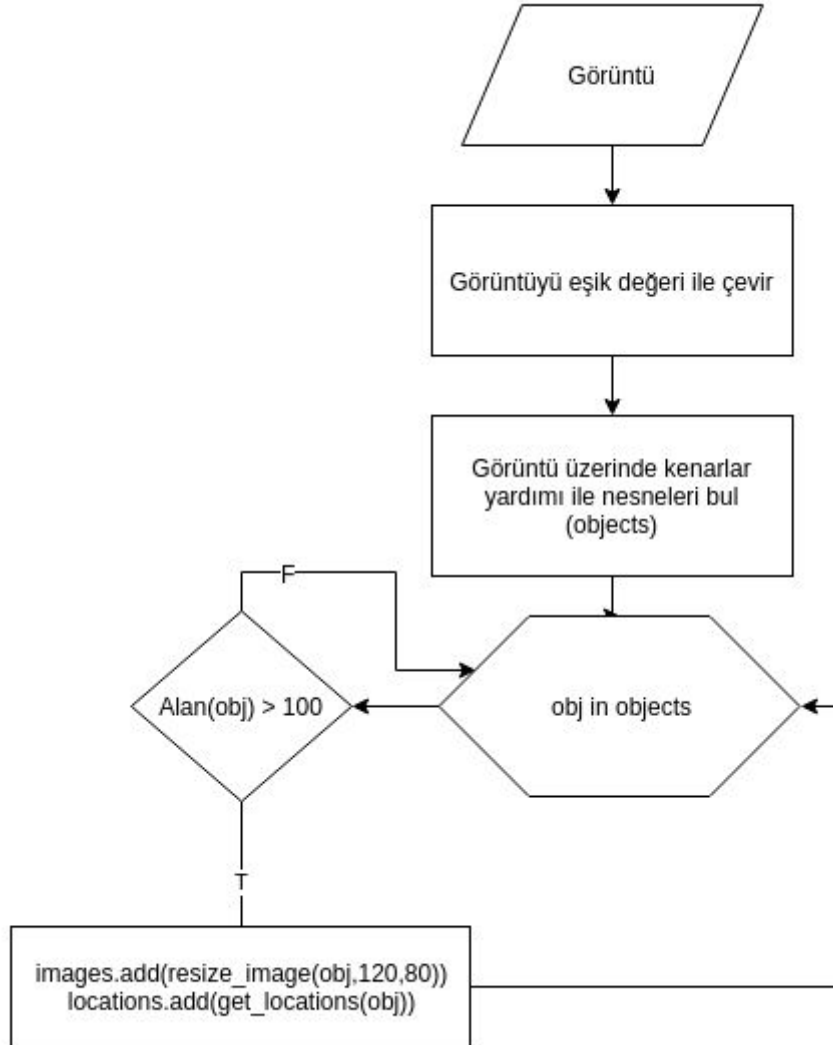
Şekil 6 Derin öğrenme modeli katmanları

Derin öğrenme modeli üzerindeki eğitimler sonucunda Şekil 6 da ki derin öğrenme modeli katmanları en iyi sonucu vermiştir.



Şekil 7 Eğitimdeki doğruluk artış ve kayıp düşüş grafikleri

### 3.2. Görüntünün İşlenmesi



Şekil 8 Görüntünün işleme akış diyagramı

Şekil 8 de ki “Alan(obj) > 100” şartının konulmasının sebebi görüntülerdeki ufak nokta veya çizgileri nesne olarak tanımaması için konulmuştur.

### **3.3. Tanınan Etiketlerin HTML Koduna Çevrilmesi**

Görüntü işlendikten sonra %60 doğruluk oranı ile tanınan nesneler ve koordinatları html koda çevirme modülüne gönderilir.

Html kodları koordinatlar ile oluşturulamayacağı için öncelikle yan yana olan etiketlerin tespit edilmesi gerekmiştir. Konumları ve tipleri tespit edilen etiketler, genişlik ve uzunlukları, çevresindeki etiketlere olan uzaklıkları görüntü üzerindeki koordinatlara bağlı olarak oluşturulur.

Yazı tipi için eğer nesnenin uzunluğu 50 den fazla ise <h1> etiketi oluşturulur.

Yazı kutusu tipi için eğer nesnenin uzunluğu 30 dan fazla ise <textarea> etiketi oluşturulur.

## BÖLÜM 4. SONUÇLAR VE ÖNERİLER

Proje de el ile çizilen mock up tasarımları görüntüsünün bilgisayar ortamına aktarılması ve html diline çevrilerek makine ortamına taşınması hedeflenmiştir. Görüntünün aktarıldıktan sonra içerisindeki nesnelerin ayrıştırılması amacıyla görüntü işleme, ayrıştırılan nesnelerin belirlenmesi amacıyla makine öğrenmesi teknolojisi bu işlemler ve html tasarımına çevrilmesi için Python yazılım dili ve kütüphaneleri kullanılmıştır.

Kullanıcı mock up araçlarını kullanarak kendi gereksinimlerine uygun web tasarımının çizimini yapmalıdır. Çizimi görüntü aktarım aracı vasıtasıyla (webcam, telefon kamerası, profesyonel kamera vb.) çizimini bilgisayar ortamına aktarmalıdır. Aktarılan tasarım görüntüsü yazılımın arayüz programına yüklenerek html tasarımına çevrilmektedir. Kullanıcı tasarım üzerinde düzenleme yapmak istediği durumlarda tekrar görüntüyü düzenleyip aktarmak yerine arayüz üzerinden ayarlamaları gerçekleştirebilmektedir. Bu ayarlar ile istenilen tasarım nesnesinin boyutları, koordinat ayarları, içeriğinde değişiklik yapabilmekte, istenildiği durumda yeni nesne ekleyebilmekte veya nesne çıkartma işlemi yapabilmektedir.

Bu çalışma ile kullanıcının el ile web sitesi tasarımlarının çizimlerini veya elektronik ortamda yaptığı web sitesi tasarım çizimlerini ayrı olarak kodlamasının önüne geçmesi, bu sayede kullanıcının tasarımdan kodlama sürecine geçiş sürecindeki zaman kaybının önlenmesi amaçlanmıştır. Kullanıcı bu projeyi kullanarak olası tasarım hatalarının düzeltilmesini; herhangi bir araştırmaya gerek duymadan projenin arayüzü üzerindeki yardımcı butonlar yardımıyla sağlayabilmektedir.

Kullanıcı html kodlarına el ile müdahale etmek istediği durumlarda programın verdiği çıktıyı açarak istediği değiştirmeyi sağlayabilir ve kodlama üzerinde oynamalar yapabilir.

## KAYNAKLAR

- [1] Chollet, Francois (2017), Deep Learning With Python (Manning)

## ÖZGEÇMİŞ

Ertan akır, 30.09.1995 de Bursa’da doğdu İlk orta ve lise eğitimi Bursa Orhaneli’de tamamladı. 2012 yılında Ahmet Necati Yılmaz Teknik Lisesi, Bilgisayar Bölümü’nden mezun oldu. 2015 yılında Sakarya Üniversitesi Bilgisayar Mühendisliği Bölümü’nü kazandı. 2018 yılında Yeni Nesil Yazılım Tic. Ltd. Şirketinde yazılım stajını yapmıştır.

## BSM 498 BİTİRME ÇALIŞMASI DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI

KONU : Görüntü İşleme ile Tasarımlar HTML Kodlarına Çevrilmesi  
ÖĞRENCİLER (Öğrenci No/AD/SOYAD): G140910054 – Ertan Çakır

Değerlendirme Konusu	İstenenler	Not Aralığı	Not
<b>Yazılı Çalışma</b>			
<b>Çalışma klavuza uygun olarak hazırlanmış mı?</b>	x	0-5	
<b>Teknik Yönden</b>			
<b>Problemin tanımı yapılmış mı?</b>	x	0-5	
Geliştirilecek yazılımın/donanımın mimarisini içeren blok şeması (yazılımlar için veri akış şeması (dfd) da olabilir) çizilerek açıklanmış mı?	x	0-7	
Blok şemadaki birimler arasındaki bilgi akışına ait model/gösterim var mı?	x	0-7	
Yazılımın gereksinim listesi oluşturulmuş mu?			
Kullanılan/kullanılması düşünülen araçlar/teknolojiler anlatılmış mı?	x	0-7	
Donanımların programlanması/konfigürasyonu için yazılım gereksinimleri belirtilmiş mi?			
UML ile modelleme yapılmış mı?	x	0-7	
Veritabanları kullanılmış ise kavramsal model çıkarılmış mı? (Varlık ilişki modeli, noSQL kavramsal modelleri v.b.)			
Projeye yönelik iş-zaman çizelgesi çıkarılarak maliyet analizi yapılmış mı?			
Donanım bileşenlerinin maliyet analizi (prototip-adetli seri üretim vb.) çıkarılmış mı?			
Donanım için gerekli enerji analizi (minimum-uyku-aktif-maksimum) yapılmış mı?			
Grup çalışmalarında grup üyelerinin görev tanımları verilmiş mi (iş-zaman çizelgesinde belirtilebilir)?			
Sürüm denetim sistemi (Version Control System; Git, Subversion v.s.) kullanılmış mı?	x	0-7	
Sistemin genel testi için uygulanan metotlar ve iyileştirme süreçlerinin dökümü verilmiş mi?			
Yazılımın sızma testi yapılmış mı?			
Performans testi yapılmış mı?			
Tasarımın uygulamasında ortaya çıkan uyumsuzluklar ve aksaklıklar belirtilerek çözüm yöntemleri tartışılmış mı?			
<b>Yapılan işlerin zorluk derecesi?</b>	x	0-25	
<b>Sözlü Sınav</b>			
<b>Yapılan sunum başarılı mı?</b>	x	0-5	
<b>Soruları yanıtlama yetkinliği?</b>	x	0-20	
<b>Devam Durumu</b>			
<b>Öğrenci dönem içerisindeki raporlarını düzenli olarak hazırladı mı?</b>	x	0-5	
<b>Diğer Maddeler</b>			
<b>Toplam</b>			

DANIŞMAN (JÜRI ADINA):  
DANIŞMAN IMZASI: