

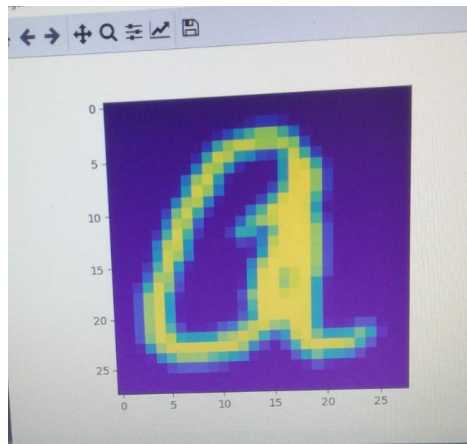
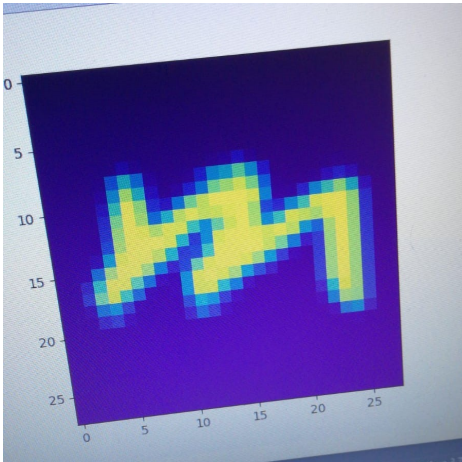
Yapay Zeka Projesi

Problemin Tanımlanması:

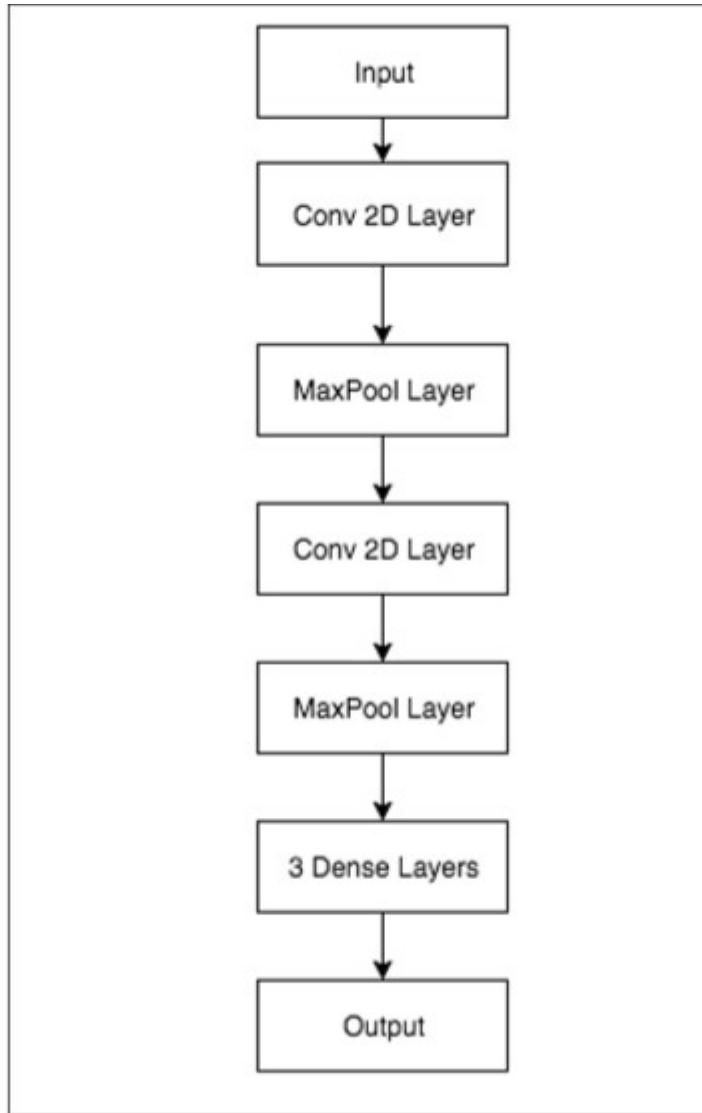
Bu proje yapay zeka ve derin öğrenme tekniklerini kullanarak el ile bir kağıt üzerine yazılan herhangi bir dökümanı resim (png, jpg vs.) formatına dönüştürerek bunu bilgisayar ve elektronik ortamda el ile yazılmış karakterleri tanıyarak bu karakterleri kelime haline dönüştüren ve dönüştürülen verilerden istenilenleri dataset (.csv) haline getiren bir algoritma yazılım projesidir. Amacı, işletme belgeleri için otomatik veri girişi sağlanması ve iş yükünden, zamandan kazanç sağlanmaktır. Konvolüsyonel Sinir Ağı (CNN) yaklaşımıyla -verilen çek senaryosu için - geliştirdiğim kelime tanıma sistemidir.

Çözümler ve Tasarım:

Tanımladığımız problem çerçevesinde öncelikle düşündüğüm ilk nokta el yazısı karakterlerini barındıran bir dataset olup olmadığı idi. Araştırmalarım sonucunda ilk bulduğum dataset mnist dataset oldu fakat derinlemesine yaptığım araştırmada mnist datasetin 60.000 -elle yazılan- rakamı tanıdığını gördüm. Bu yüzden daha gelişmiş versiyonu olan Emnist datasetin 697.932 karakter tanımlı hem rakam hem de harfleri tanıma kabiliyeti olduğundan bu dataseti kullanmaya karar verdim. Daha sonrasında bu dataset ile uyumlu çalışabilecek Python'da yapay zeka kütüphaneleri araştırmaya başladım. İnternet üzerinden incelediğim örnekler sonucunda Tensorflow ve Keras Kütüphanelerini kullandım. Bu adımlardan sonra ilk hedefim okuduğum datasetten karakterler elde etmektir. Bunun için datasetten elde edilen verileri Python'nun Numpy Kütüphanesini kullanarak verileri dizili dizilere dönüştürdüm ve görüntüleri aralıklara göre normalleştirdim. Sonrasında ön işleme için tüm görüntüleri 28*28 biçiminde yeniden biçimlendirdim. Daha sonra bahsedeceğimiz "Model" oluşturmak için öğrenme ve test verilerini yeniden şekillendirdim. Sonrasında Python Matplotlib Kütüphanesini kullanarak bu karakterleri görsellere çevirdim.

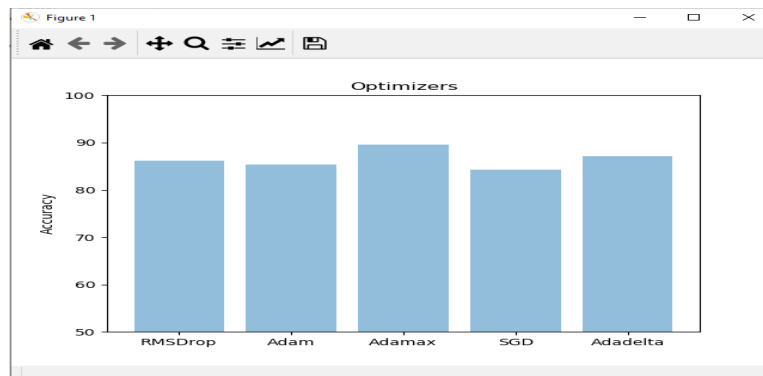
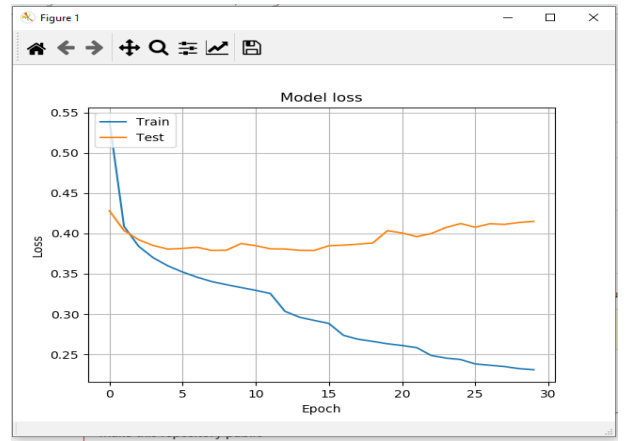
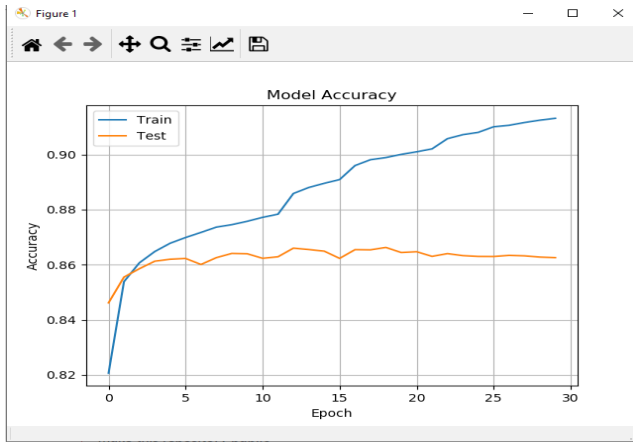


Bu aşamadan sonra model oluşturmam gerekti. 28*28 biçiminde yeniden biçimlendirdiğim karakterlerle bir model oluşturdum. Bu model aşağıdaki şekilde görüldüğü gibi 2 Convolution ve 3 Dense katmanlarına sahiptir. İlk Convolution katmanı 5*5 boyutunda bir çekirdek kullanır ve 32 filtreye sahiptir. İkincisi ise 5*5 boyutunda bir başka çekirdek ve yine 32 filtre kullanır. Relu aktivasyon işlevini kullandım. Her Convolution tabakasından sonra 2x2 havuz boyutuna sahip maxpooling katmanlarımız var.



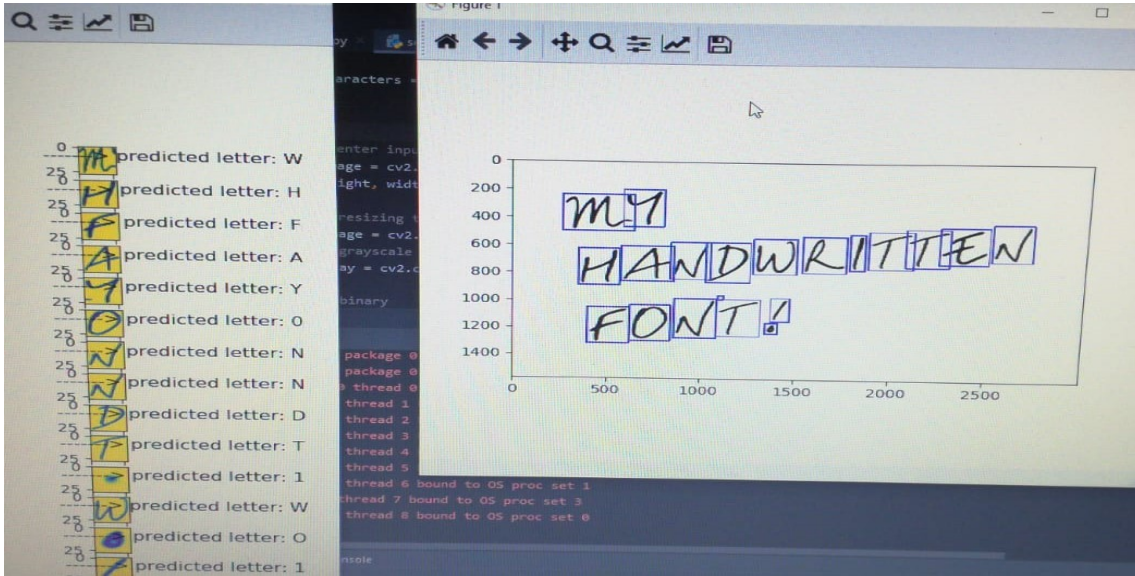
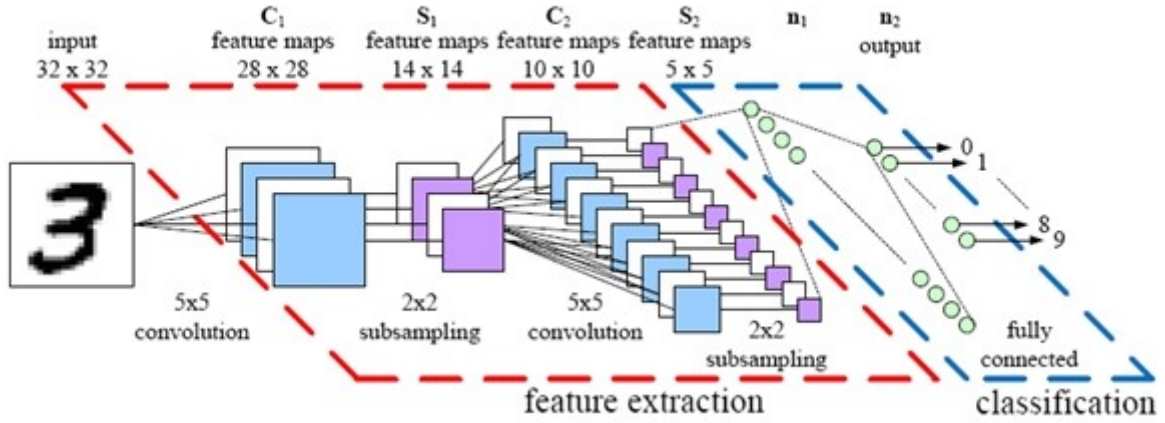
Daha sonra Emnist datasetinden elde edilen eğitim ve test meteryallerini 30 epochs yani 30 adımda eğitim yaptırđım. Elde edilen model içindeki veri kümesinde 62 sınıf, 26 büyük harf, 26 küçük harf ve 10 rakam vardır. Bu veri kümesinde %86.33'lük bir doğruluk elde ettim.

```
697664/697932 [=====>.] - ETA: 0s - loss: 0.2288 - acc: 0.9137
697728/697932 [=====>.] - ETA: 0s - loss: 0.2288 - acc: 0.9137
697792/697932 [=====>.] - ETA: 0s - loss: 0.2288 - acc: 0.9137
697856/697932 [=====>.] - ETA: 0s - loss: 0.2288 - acc: 0.9137
697920/697932 [=====>.] - ETA: 0s - loss: 0.2288 - acc: 0.9137
697932/697932 [=====] - 2364s 3ms/step - loss: 0.2288 - acc: 0.9137 - val_loss: 0.4128 - val_acc: 0.8631
Accuracy: 86.33%
Model Test
dict_keys(['val_loss', 'val_acc', 'loss', 'acc', 'lr'])
```



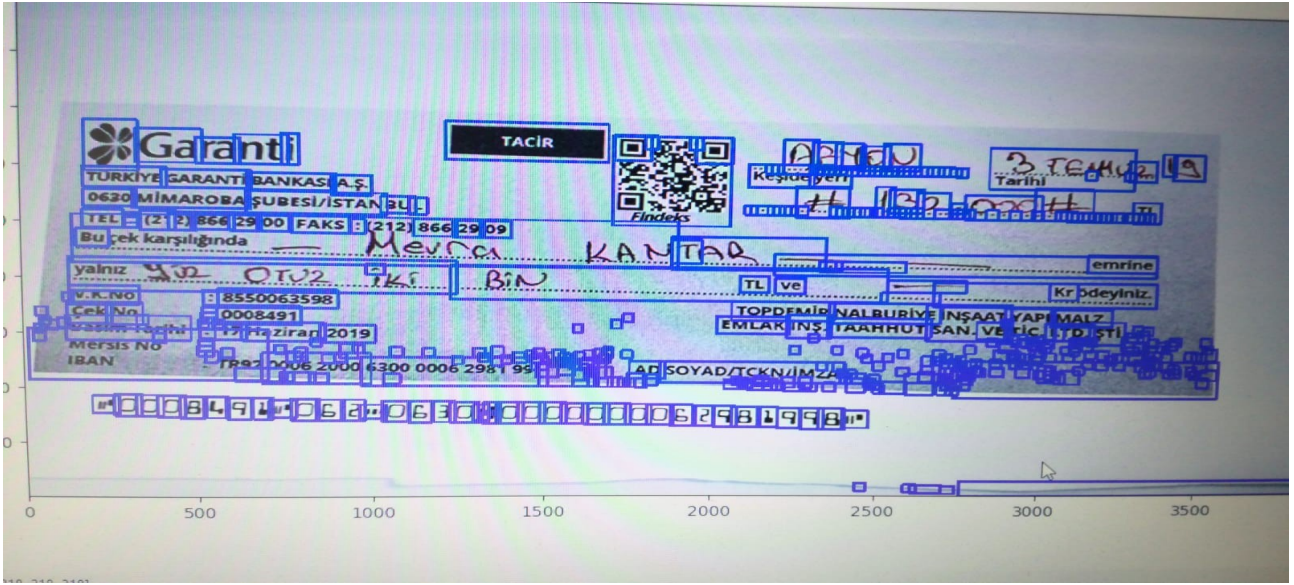
Olusturduğum modelden de teorik olarak bahsetmeden geçmeyelim. CNN diye projemizin probleminde geçen terimin açılımı "Konvolüsyonel Sinir Ağı" dır. Günümüzde görüntü işleme ve görüntü tanımlama alanında önem bir rol oynamaktadır. Günümüzde özellikle bilgisayar görmesinde, resimlerin sınıflandırılmasında, akıllı araçlarda (otomatik pilot tekniklerinde),

trafik ışıkları, askeri projelerde nesne tanımlama vs. aktif kullanılmaktadır. Aşağıda bulunan CNN'yi anlatan bu görseli temsili olarak koyuyorum.

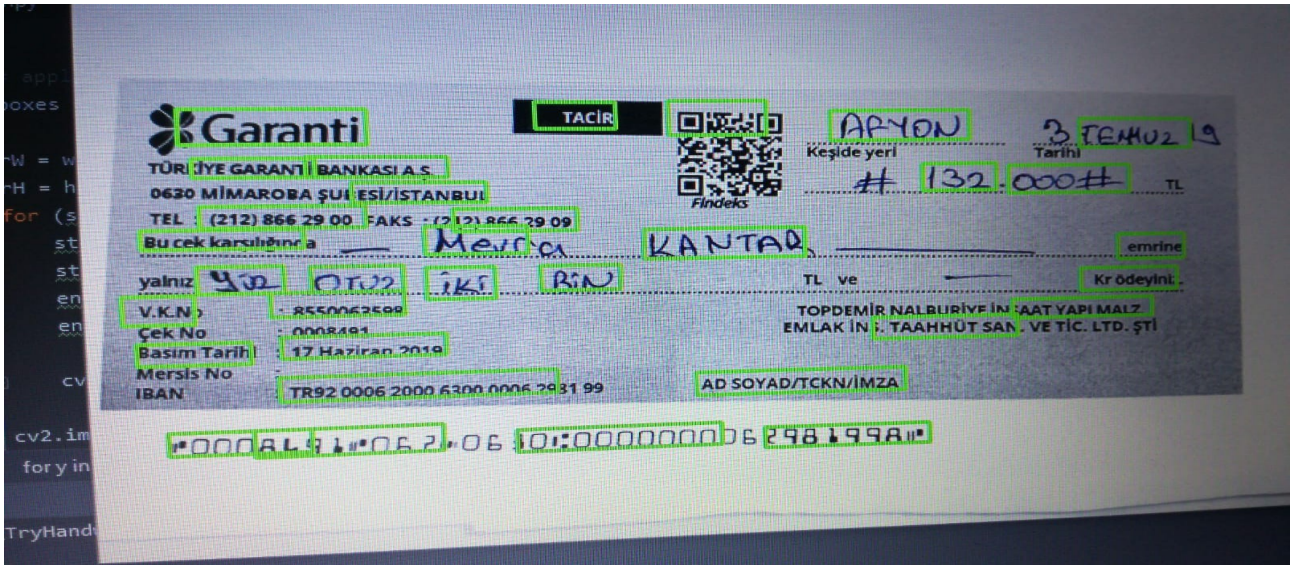


Yukarıdaki resim ilk tahmin görüntüsü örnek olması için koyuyorum

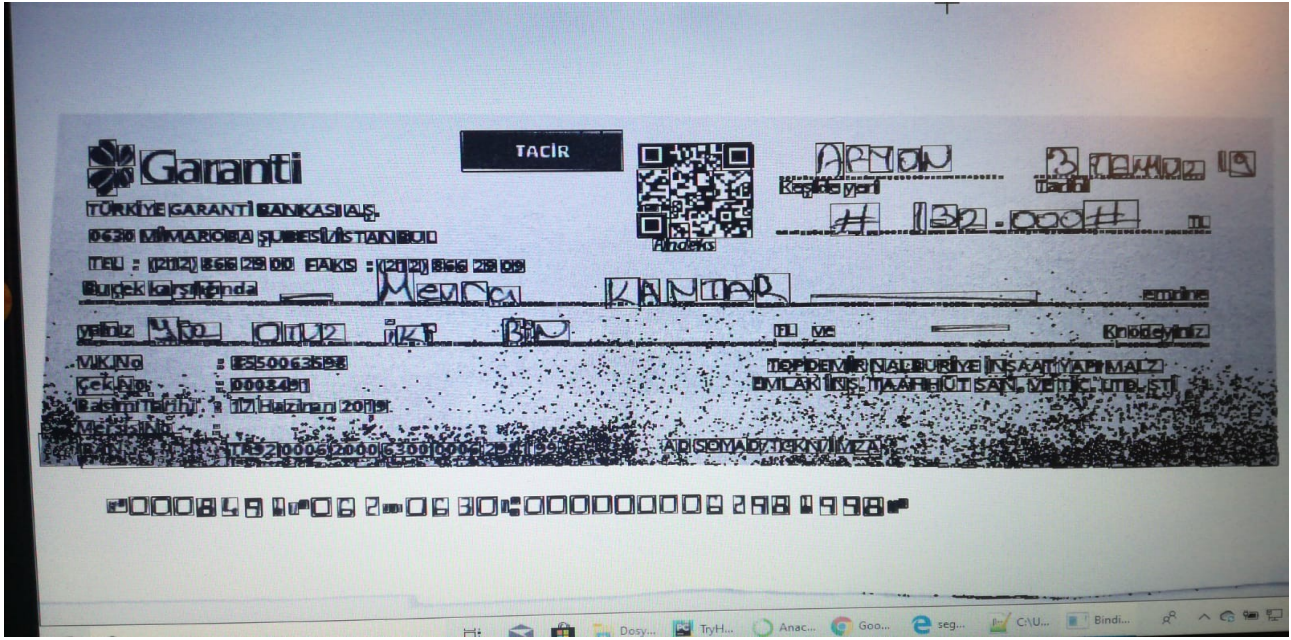
Modellimizi oluşturduk bundan sonraki en büyük problemimiz bize verilen çek üzerindeki karakterleri yakalayabilmektir. Tabiki de en önemli görüntü işleme kütüphanesi Open cv kullandım. Burdaki en büyük problem bize verilen çek resimleri dijital ortamda oluşturulmadığı için (Tarayıcı vs. çıkmadığından), resim negatif forma dönüştürüldüğünde sanki bütün terimler el yazısı gibi algılanıyordu. Aşağıdaki resimlerdeki gibi.



Daha sonrasında kelimeleri çek üzerinden keserek bunları sırası ile işlemeye çalıştım. Bu işlemde daha önce her karaktere verdiğimiz 28*28 (w,h) boyutu yüzünden kesilen kelimelerin tekrar okunması sırasında resim boyutu problemi ortaya çıktı.



Bu yöntemde tam bir çözüm olmadığından dolayı sancılı bir araştırma ve düşünme sürecinden sonra çek üzerindeki bütün karakterlerin tam boyutunu tespit edip tek bir fonksiyonda hiyerarşik olarak tespit edilen karakterleri kırparak bu katakterleri aşağıda gösterilen resimdeki gibi tahmin ettirdim



Bu yöntemle metni harflere ayırdım. OpenCV’de kullanılan yöntemler; findContours ,threshold, ve resmin hatalarını bulmak için “erode ” yöntemlerini kullandım. Aslında açıkça yaptığım işlem sırasıyla harflerin genel ana hatlarını, genel hatlarını sonra iç hatlarını bulmaktı. Bu durumda bize de sadece harflerin ana hatları gerekiyordu. Daha sonra önceden bahsettiğim gibi hiyerarşik bir ağaç elde ediyoruz (cv2.RETR_TREE parametresi). Bir sonraki adım ise 28x28 kareye ölçeklendirilen her bir harfi kaydetmekti. Daha öncede numpy temelinde üretmiş olduğum veri kümelerininikulandım. Bu veri kümeleri ile bir tahmin yaptırımdım ama burada da bir sorun ile karşılaştım. Karakterler tek tek tahmin edildiği için anlamlı bir kelime elde edemiyordum. İşte bu ihtiyaçtan ötürü kırpma ve ölçekleme için dizilerle çalışma fonksiyonlarını kullandım. Elde ettiğim karakterlerden nasıl anlamlı bir veri oluştururum derken bir örnek ile karşılaştım ve beni fikir olarak olgunlaştırdı. Aslında gayet basit bir fikirdi, proje içinde inceleyiniz zaman göreceksiniz. Tahmin edilen harflerin arasındaki boşluk karakterlerinin $\frac{1}{4}$ ’ ünden fazlasına boşluk eklemek için önceden kaydedilen karakter genişliğini kullandım. Böylelikle anlamlı bir veri elde ettim. Resimden kaynaklanan istenmeyen alanları belirleyip tahmin ettirdim. Elde edilen string ifadeden verilen boşluklar sayesinde istenmeyen verilerden kurtulup data.csv dosyasına istenilen veriler kaydedildi. söylemiştim. Elde edilen string ifadeden verilen boşluklar sayesinde

istenmeyen verilerden kurtulup data.csv dosyasına istenilen veriler kaydedildi.

Kullanılan Yazılım Araçları Kütüphaneler:

- TensorFlow
- Keras
- Numpy
- Matplotlib
- Emnist Dataset (Emnist.loader)
- OpenCv (python - cv2)

Riskler ve Projenin gelişim önerileri:

Öncelikle bütün yapay zeka projelerinde özellikle görüntü işleme konusunda çok iyi donanıma sahip bir bilgisayara ihtiyacımız var. Ben bu proje de en çok bunun sıkıntısını yaşadım. Kısacası daha iyi bir makine ile oluşturduğumuz modele daha çok eğitim yaptırıldığında başarı oranı daha yüksek ve daha hızlı bir sonuç elde edilecektir. Genelde tahminlerin içinde yanlış tahmin edilen karakterler olmaktadır. Bunun için bir isim database oluşturulup bu database içindeki en yakın ismi string olarak döndürülebilir. Tahmin edilen tarih yanlışsa bunu Python da düzelten kütüphaneler mevcut bunlar da kullanılabilir. Bir başka konu ise, süre kısıtlamasından dolayı en basit ve en stabil çözümü sunmaya çalışmak oldu. “Sklearn” gibi kütüphaneler kullanarak farklı yöntemler kullanılabilir.

