Çevrimiçi Eğitimde/Sınavda Kullanılacak Yüz Tanıma Sistemi

- Yüz tespit ve tanıma işlemlerinin hızlı olması gerekmekte.
- Az sayıda yüz görseli ile tanıma yapabilmesi lazım.

Yüz Tespiti

- Viola-Jones benzeri bir gerçekleme yapıldı.
- Haar-Benzeri öznitelikler kullanıldı.
- Özellikleri seçme işlemi için AdaBoost yerine Random Forest kullanıldı.
- Sci-kit learn, LFW yüz veri seti.

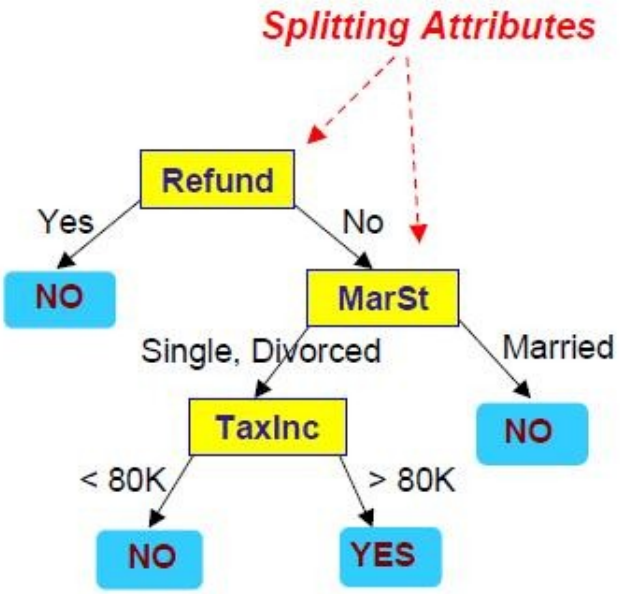
Random Forest

- Çok sayıda karar ağacından (decision tree) oluşur.
- Her karar ağacı modelin yaptığı tahmine katkıda bulunur.
- Ağaçların tekil olarak yaptığı tahminde/hatada diğer ağaçlar olarak korelasyonun az olması beklenir.

• Karar ağacı

Tid	Refund	Marital Status	Taxable Income	Cheat
	categorical	categorical	continuous	class
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data



Model: Decision Tree

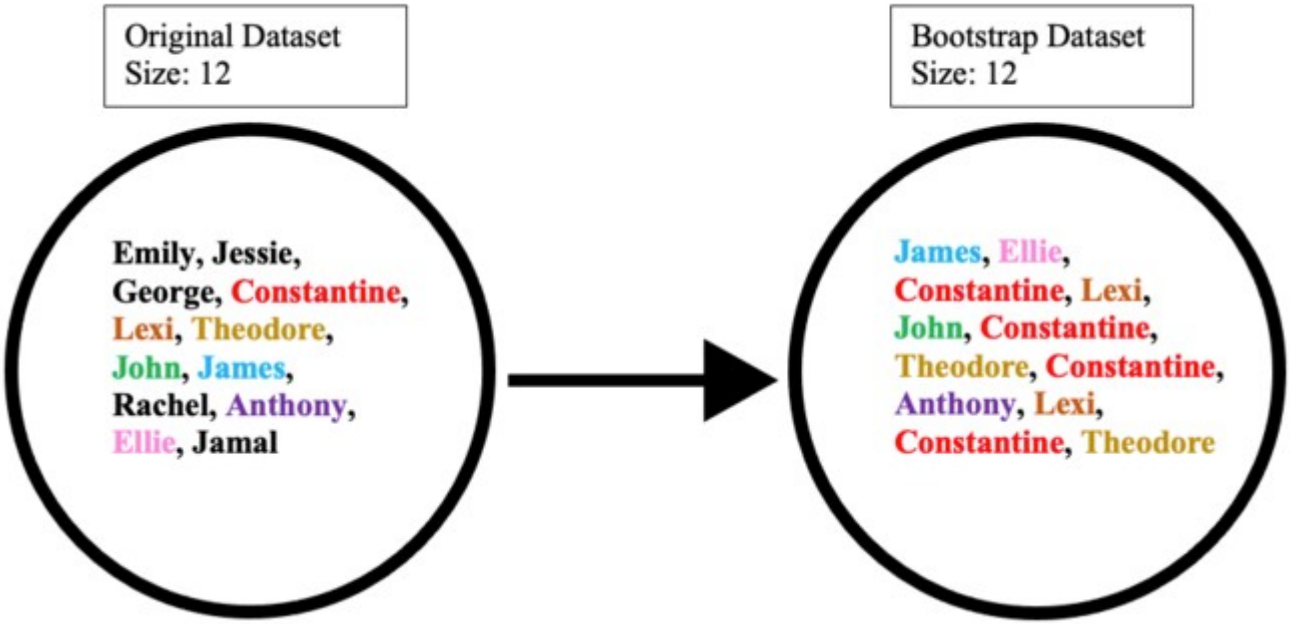
• Karar ağacının oluşturulması

- Veri setindeki tüm özelliklerinin veri setini ne kadar iyi ayırdığı ölçülür. En iyi ayıran öznitelik kök düğüm(root node) olur.
  - Gini safsızlığı, entropi (Gini impurity)
- Dalların seçimi için ise veri setinin ilgili düğümün ifade ettiği alt küme için bir önceki adımda anlatılan işlem tekrarlanır.
- Oluşturulacak diğer tüm düğümler için bu işlem devam eder.
- Over-fitting

Random Forest – Karar Ağacı Oluşturulması

- Bootstrap aggregating (Bagging)
  - Verisetindeki M adet veriden N adedi rastgele şekilde seçilir
  - M < N
  - Tekrarlar olabilir.

- Bootstrap aggregating (Bagging)



Rastgele Özellik Altuzayı

- Karar ağacı oluşma aşamasında bir sonraki tüğümün seçilebileceği öznitelikler mevcut tüm özniteliklerden, daha önce belirlenmiş bir oranda kısıtlanır.

Scikit – Learn Random Forest

- sklearn.ensemble.RandomForestClassifier
  - n_estimators,int, default=100 – ağaç sayısı
  - criterion{“gini”, “entropy”}, - öznitelik seçim kriteri
  - max_depth, default=None – ağacın derinliği
  - min_samples_split default = 2 – minimum dallanma sayısı

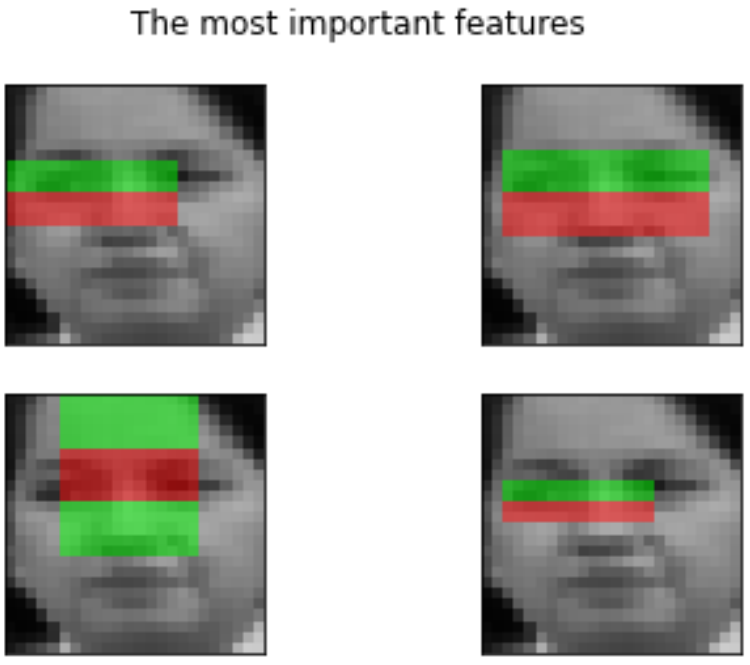
Modelin Eğitilmesi

- LFW dataseti ve yüz dışı rastgele görünütlerden oluşan bir veri seti ile eğitildi.
- 6000 Negatif, 2300 Pozitif
- Görseller 24x24x1 çöznürlüğüne dönüştürüldü.
- 24x24x1 boyutundaki NxM görselde bulunabilecek tüm 161864 adet Haar-Benzeri öznitelikler tüm görseller için hesaplandı.

Modelin Eğitilmesi

- (N+M)*161864 boyutlu öznitelik dizisi ve (N+1)*1 sonuç dizisi ile RFC oluşturuldu.
- Eğitilen modeldeki en önemli 2289 adet öznitelik alındı.
- Sadece 2289 öznitelik ile aynı şekilde yeni bir model eğitildi.
- Eğitilen yeni model kaydedildi.
- Model test setinde %97 başarıma ulaştı.

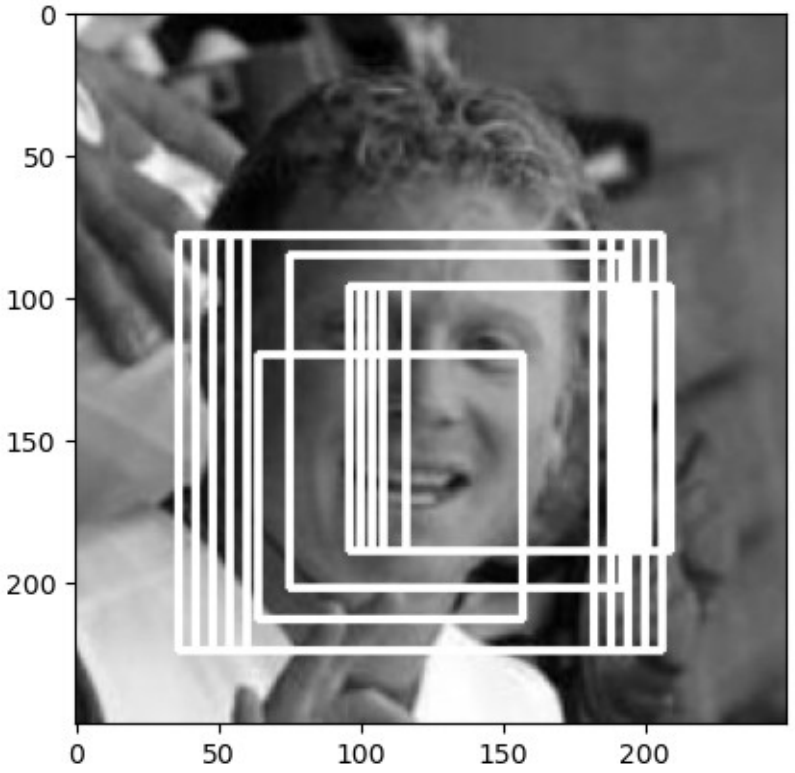
Önemli Haar-Benzeri Öznitelikler

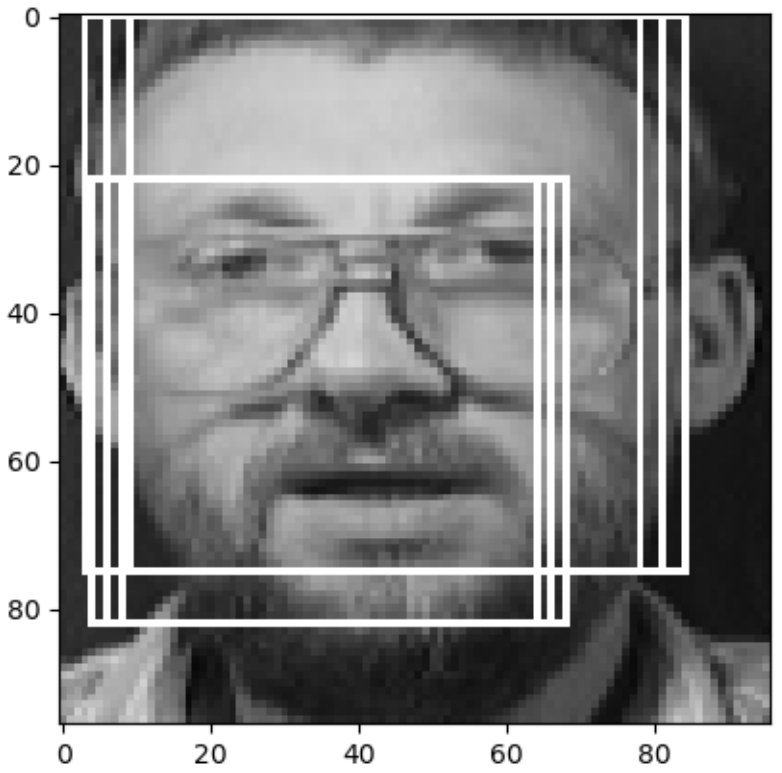


- 800 adet negatif 800 adet pozitif görsel ile eğitilmiş rastgele sınıflandırıcısındaki ağaçlarda en çok kullanılmış 4 öznitelik.

Oluşturulan Modelin Uygulanması

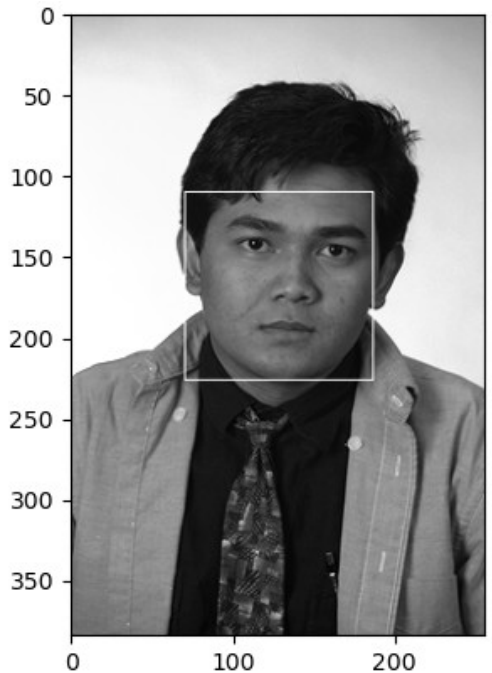
- 24x24x1 Boyutundaki görseller için eğitildi.
- Haar-Benzeri öznitelikler ölçeklenebilir.
- Herhangi bir görselin tüm noktalarına farklı ölçeklerde uygulanarak, uygulandığı pencerenin içerinde yüz varlığı test edilebilir.
- Makalede önerilen ölçek faktörü 1.25 ve pencerinin hareket miktarı ise 1 pikseldir.

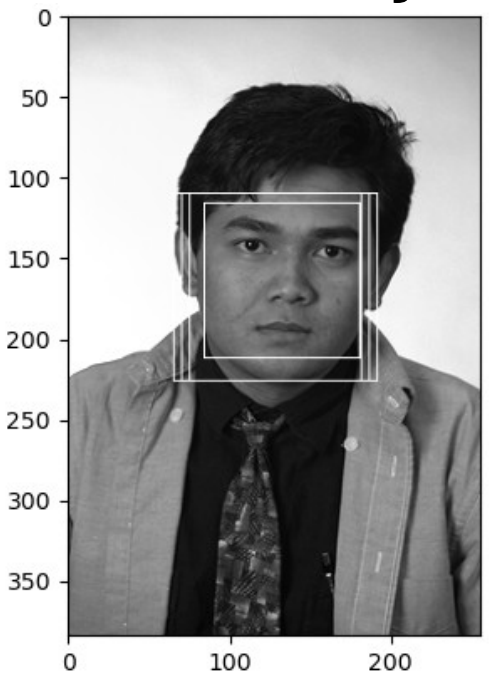

0
50
100
150
200
0
50
100
150
200

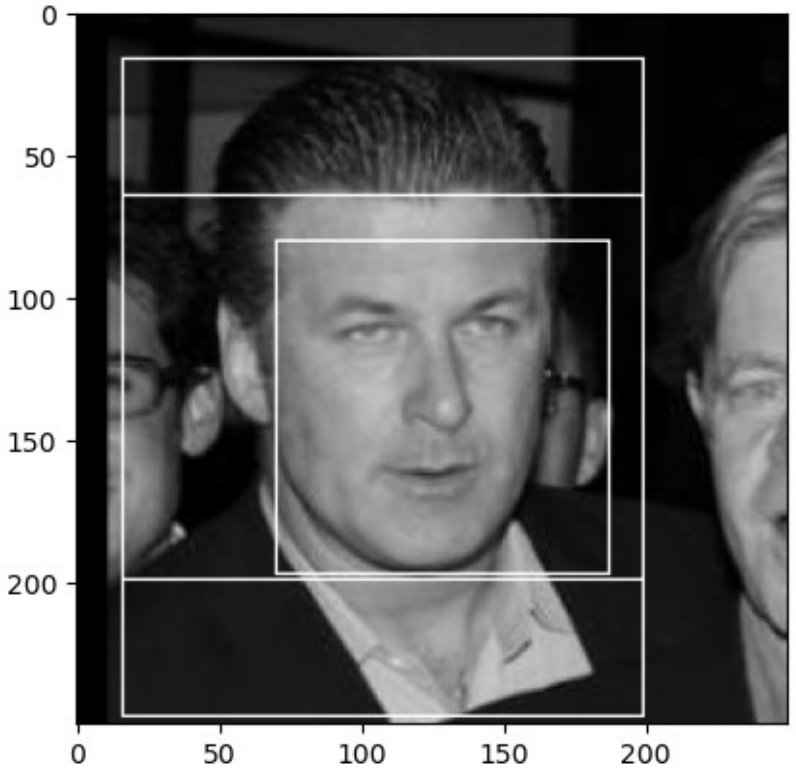

0
20
40
60
80
0
20
40
60
80

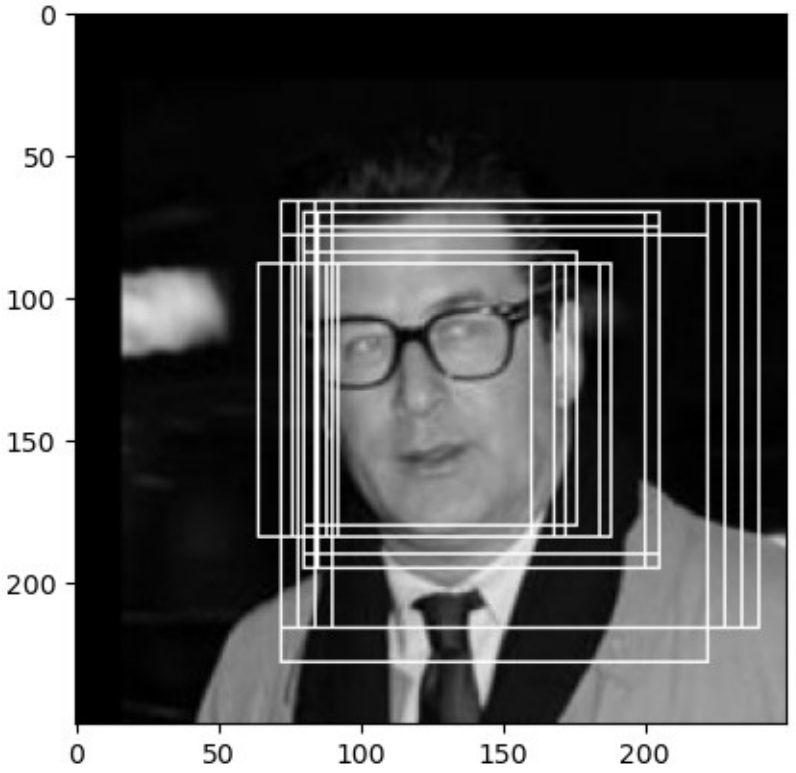
Sonuçların Filtrelenmesi

- Yüzün bulunduğu her çerçeveye yakın çerçeveler arasında en kuvvetlisi seçildi.






0
50
100
150
200
0
50
100
150
200

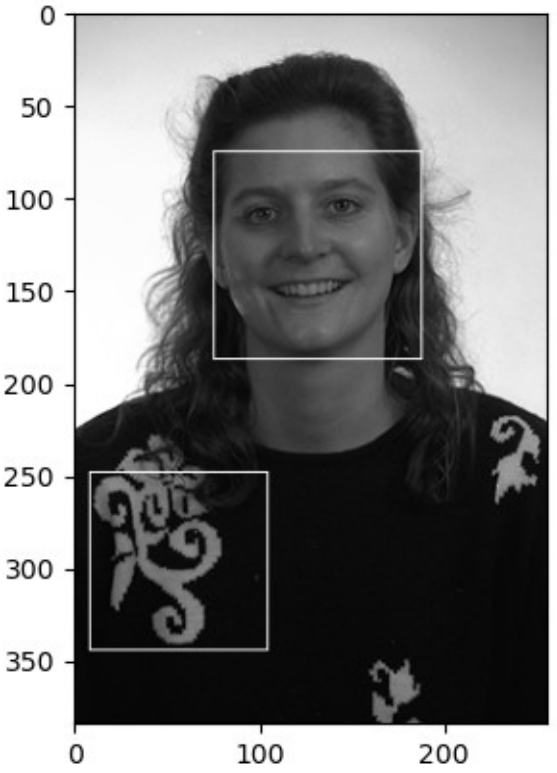

0
50
100
150
200
0
50
100
150
200

Uygulamada ortaya çıkan sorunlar ve olası çözümler

- Yavaş.
  - Özniteliklerin hesaplanması çok vakit almakta
    - Düşük boyutlu ölçekler yüz tanıma elverişsiz. Dolayısıyla hesaplanmayabilir.
    - 24x24 – 96x96
    - Memoization, farklı ölçeklerde ve konumlarda aynı Haar öznitelikleri hesaplanmakta.
    - Kaskat oluşturulabilir.
    - Haar kaskatının hesaplanması tek thread ile CPU’da çalışmakta. Kolaylıkla çok thread ile çalışacak hale getirilebilir.
    - ...

Uygulamada ortaya çıkan sorunlar ve olası çözümler

- Hatalı pozitif sonuçlar üretebiliyor.
  - Daha fazla negatif görsel ile eğitilebilir.
- Çerçeve kare olduğu için dar görsellerde hatalı sonuçlar elde edilebilmekte.
  - Görsel kare hale getirilebilir.

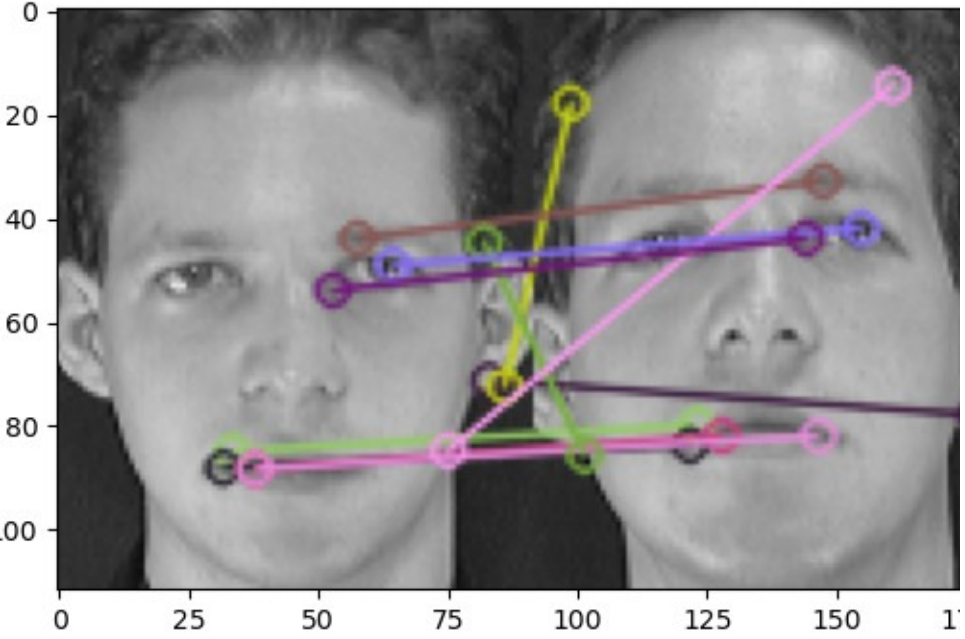

0
50
100
150
200
250
300
350
0
100
200

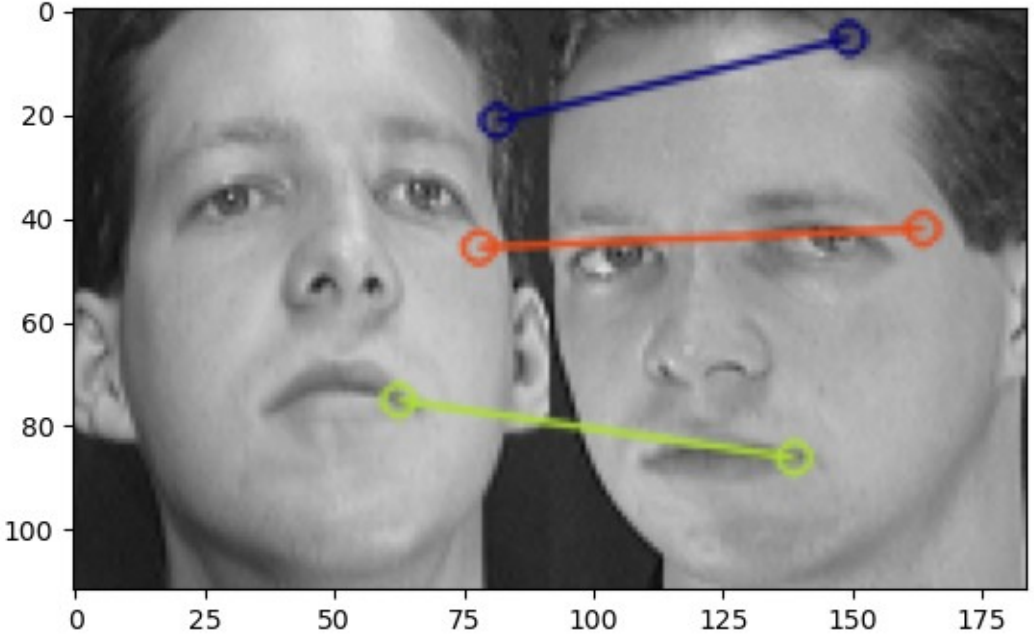
Yüz Tanıma

- SIFT ile iyi sonuçlar alınamadı.
- Siyam ağları ve VGG16 kullanılarak gerçekleme yapıldı.
- LFW, CASIA WebFace, Color FERET veritabanı

Neden SIFT Değil

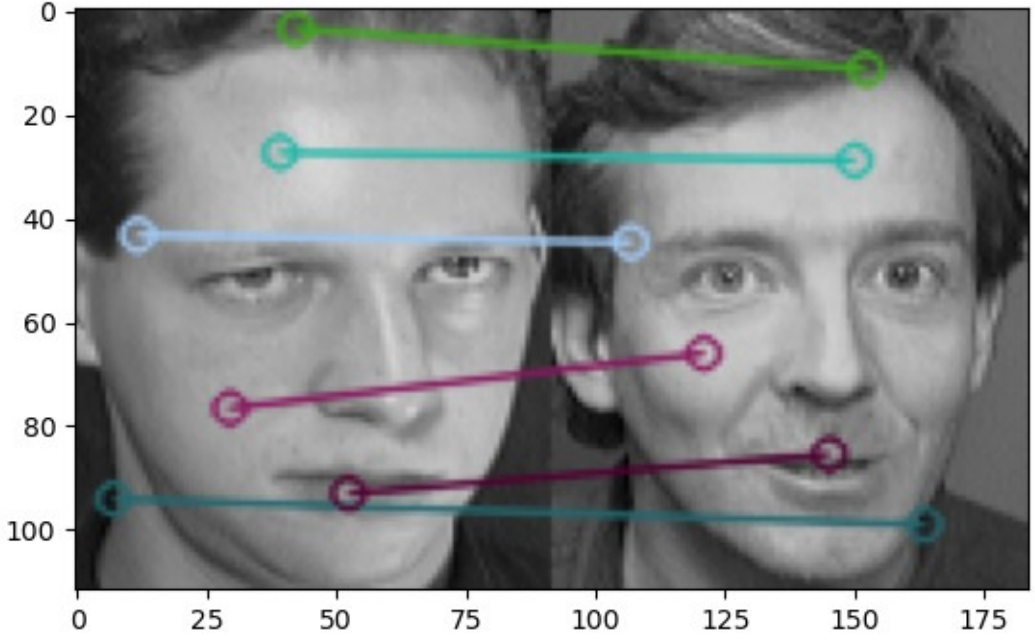
- Pozdan, mimiklerden çok ciddi etkilenmekte

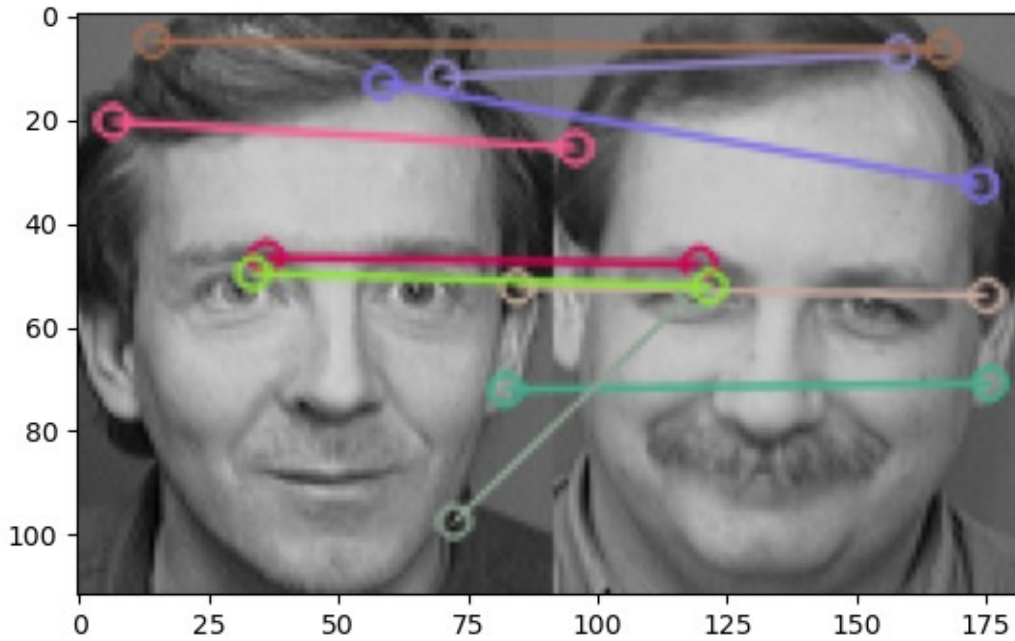




Neden SIFT Değil

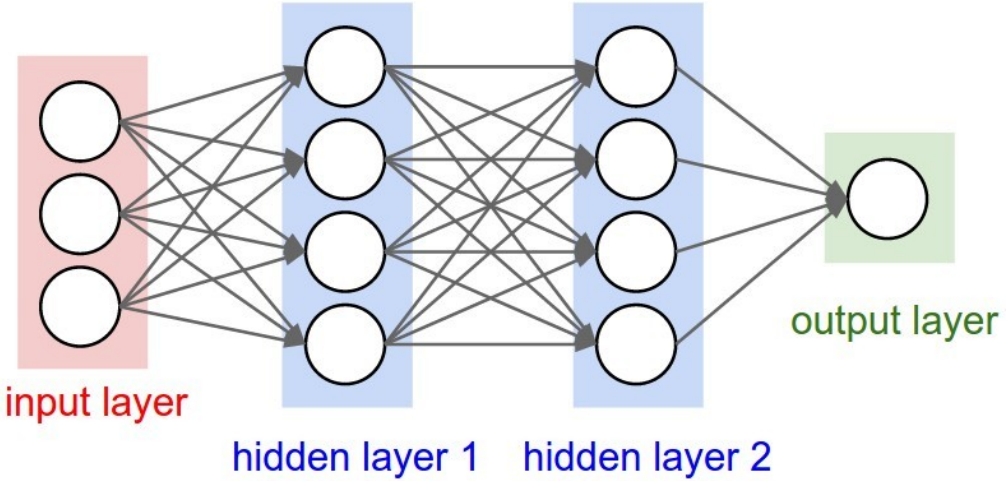
- Benzer bölgelerde çıkan yerel özellikler hatalı sonuç üretebilmekte.
- Ancak oldukça kısıtlanmış açı, poz ve ölçekte elde edilmiş görüntüler ile iyi sonuçlar elde edilebilir.





Sıradan Sinir Ağları

- Girişi olan bir vektörü aradaki gizli katmanlardan geçirerek çıkış olan vektörü elde eder.
- Giriş ve çıkış dahil her katman nöronlardan oluşmuştur.
- Her katmandaki nöron birbirinden bağımsızdır ve bir önceki ve bir sonraki katmandaki tüm nöronlara bağlıdır.



Sıradan Sinir Ağları

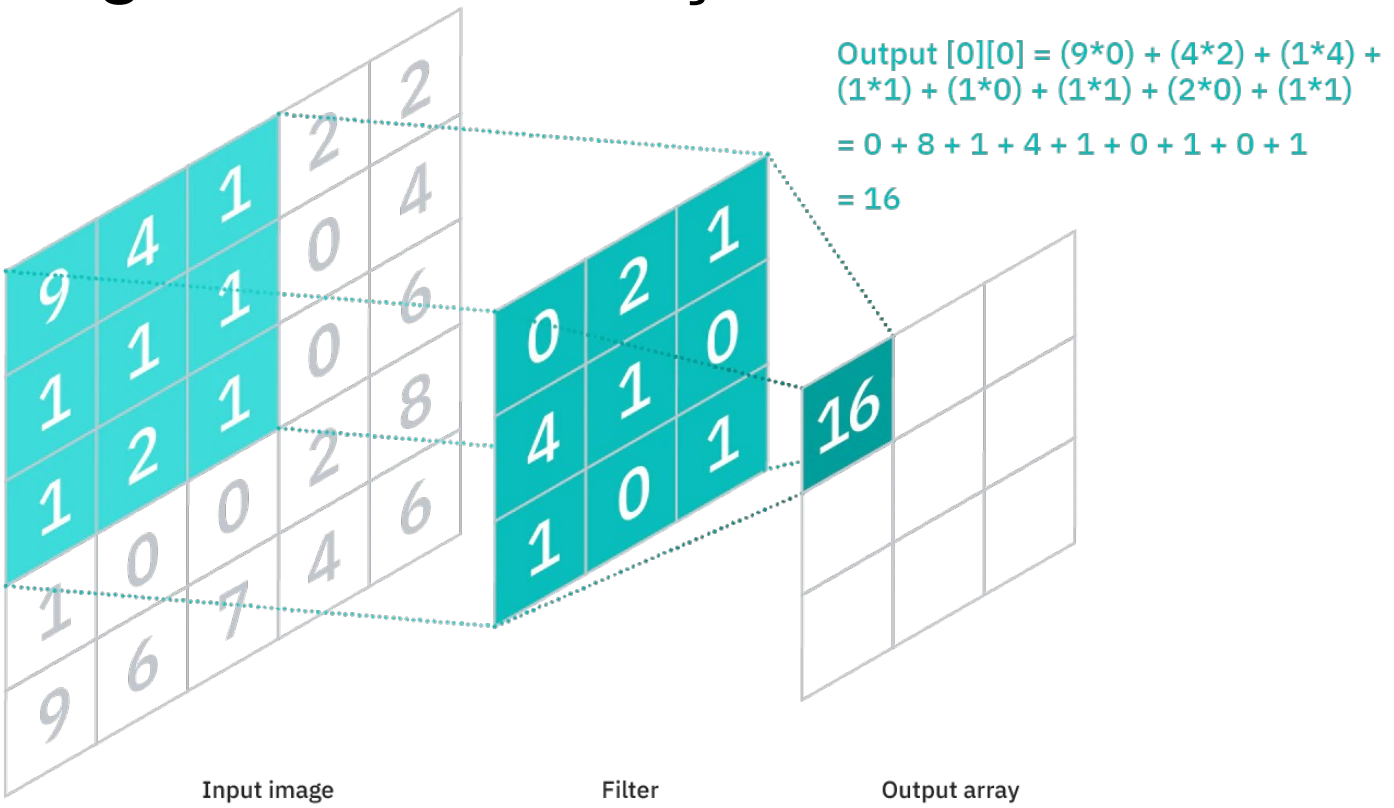
- Görseller ile kullanmak için pek uygun değillerdir.
- Örnek olarak 244x244x3 boyutunda bir görsel için giriş katmanında 178608 nöron bir sonraki gizli katmandaki tam bağlı her bir nöron için ise 178608 ağırlığa ihtiyaç vardır.
- 100 Nörondan oluşan sadece tek bir katmanda optimize edilmesi gereken 178608000 parametre var.
- Bu çok fazla sayıdaki parametre over-fitting'e dolayısıyla daha fazla kaliteli eğitim verisine ihtiyaca sebep olabilir.

Konvolüsyonel Sinir Ağları

- CONV
- FC
- RELU
- POOL

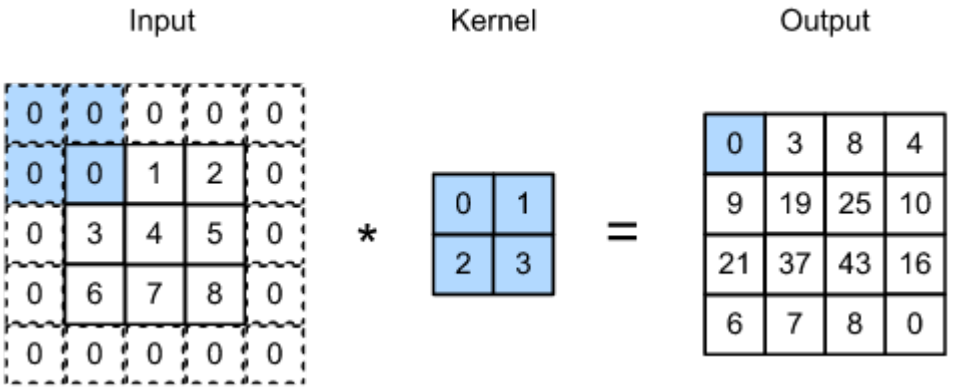
Konvolüsyon Katmanı

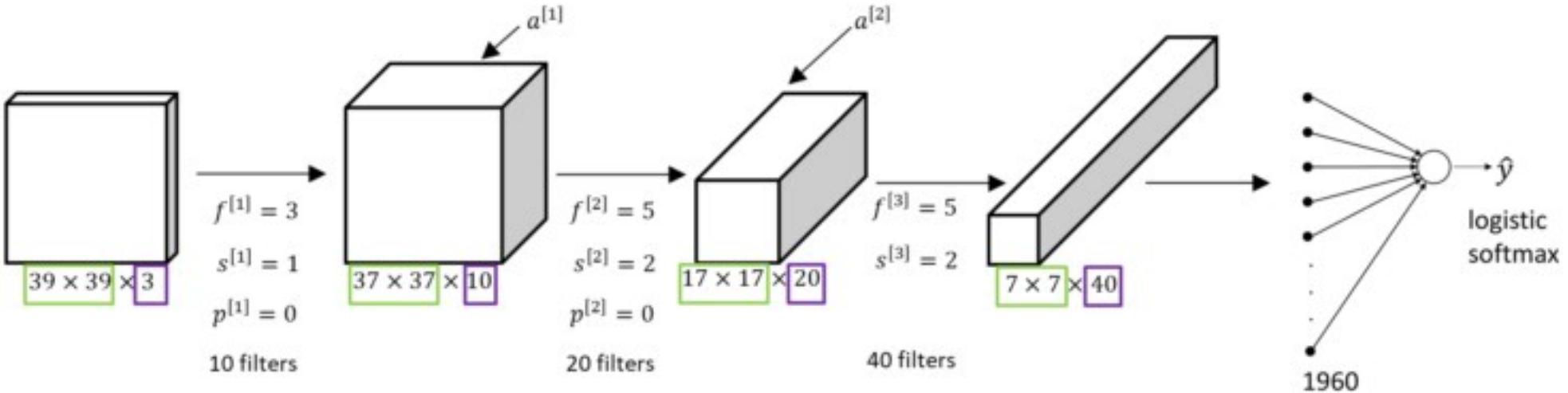
- Bir dizi öğrenebilir filtre içerir.



Padding, Stride

- (N – f + 2p)/s + 1
- Padding
  - Shrinking
  - Edges not covered by may passes





$n_H^{[0]} = n_W^{[0]} = 39$

$n_C^{[0]} = 3$

$n_H^{[1]} = n_W^{[1]} = 37$

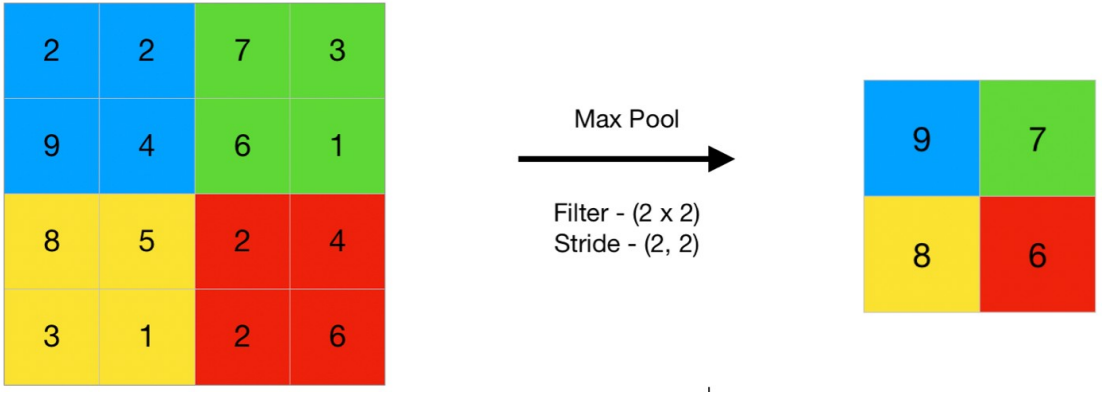
$n_C^{[1]} = 10$

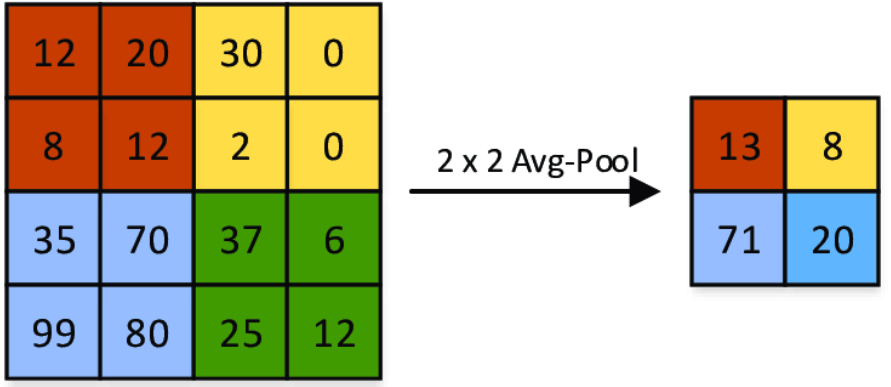
$n_H^{[2]} = n_W^{[2]} = 17$

$n_C^{[2]} = 20$

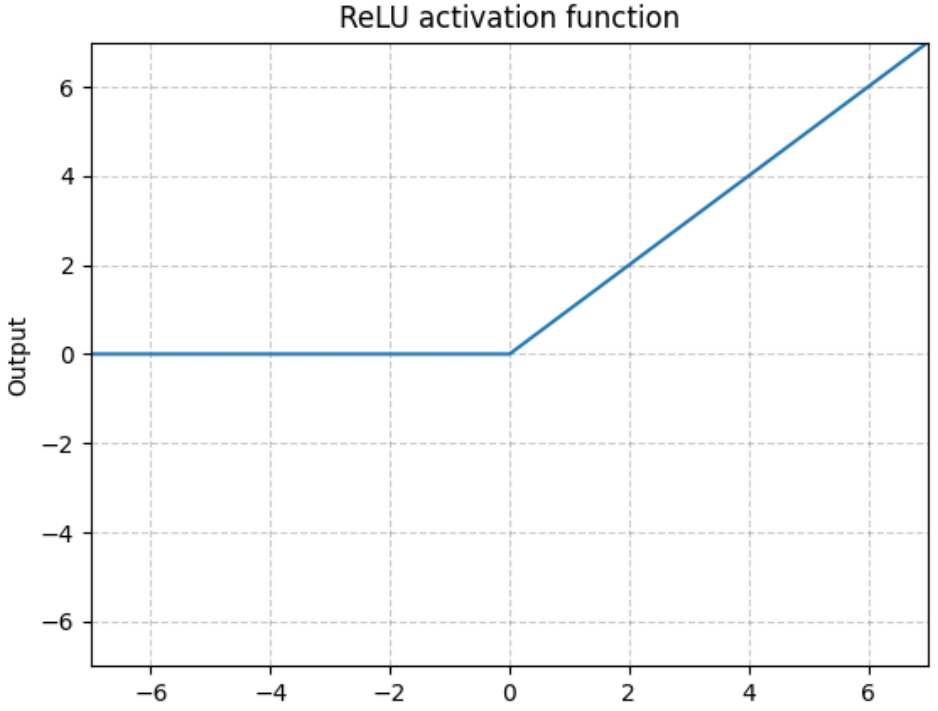
Pooling

- Öğrenme yok.
- f - filtre büyüklüğü
- s – stride
- f:2x2 s:2
- (n-f)/s + 1





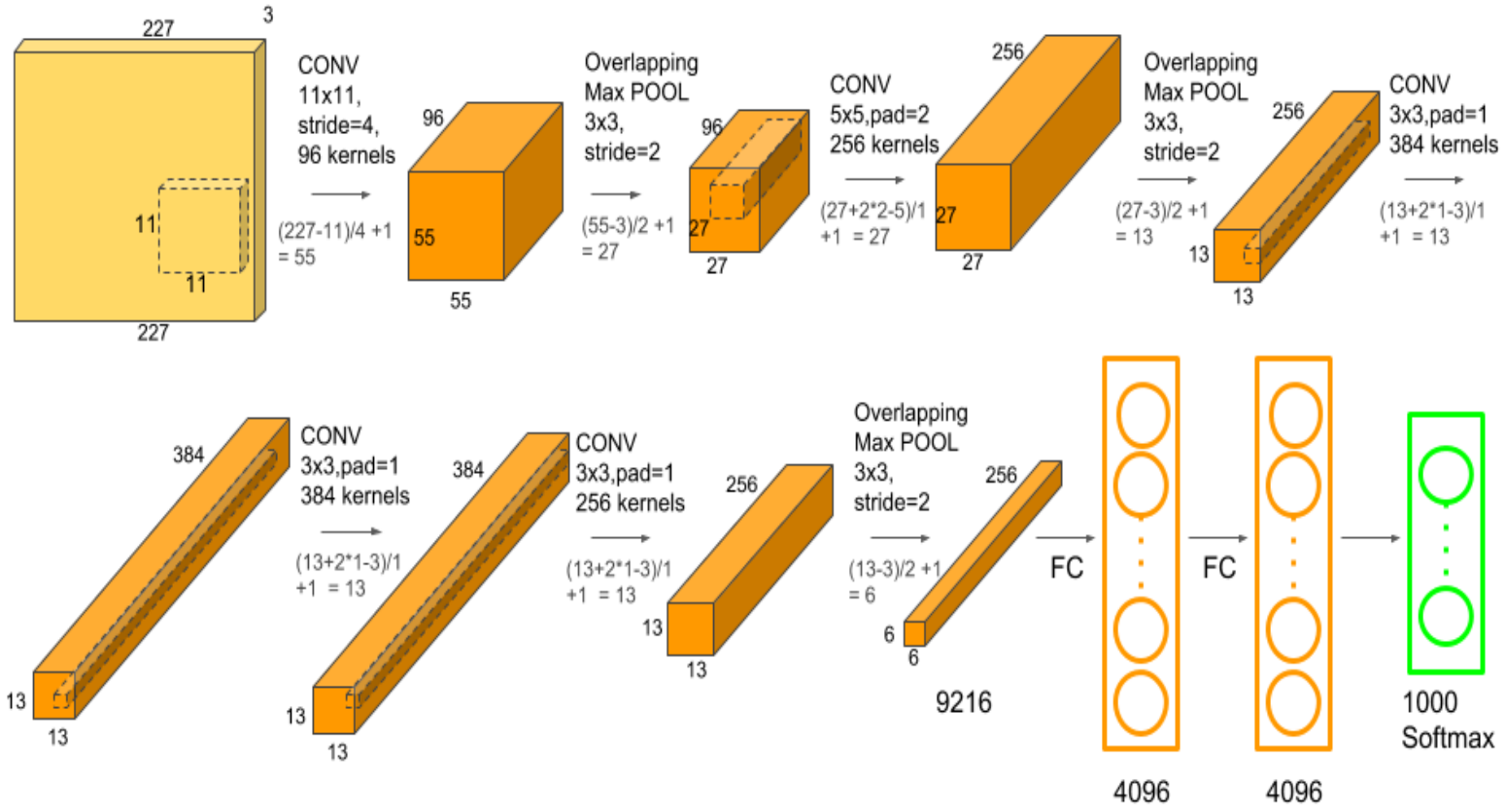
ReLU



$$\mathrm{ReLU}(x) = (x)^+ = \max(0, x)$$

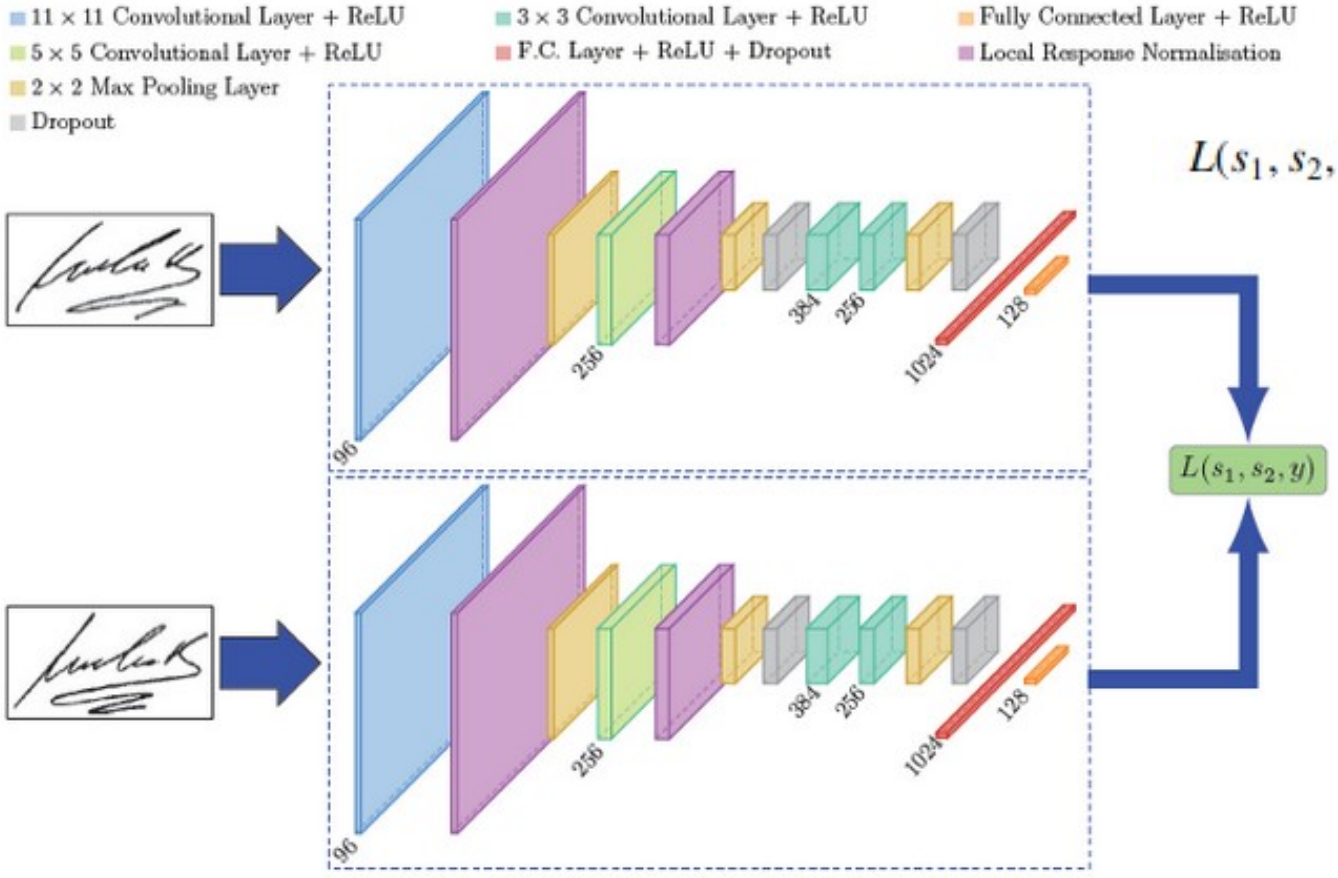
- Rectified Linear Unit Function
- Conv + ReLU + Polling

Fully Connected (FC)



Siamese Networks

- İki yada daha fazla ağı alt ağ olarak içerir.
- Ağ üzerindeki tüm ağırlılklar paylaşılır.
- İsimlendirilmiş görüntüler arasındaki benzerliği öğrenilir.

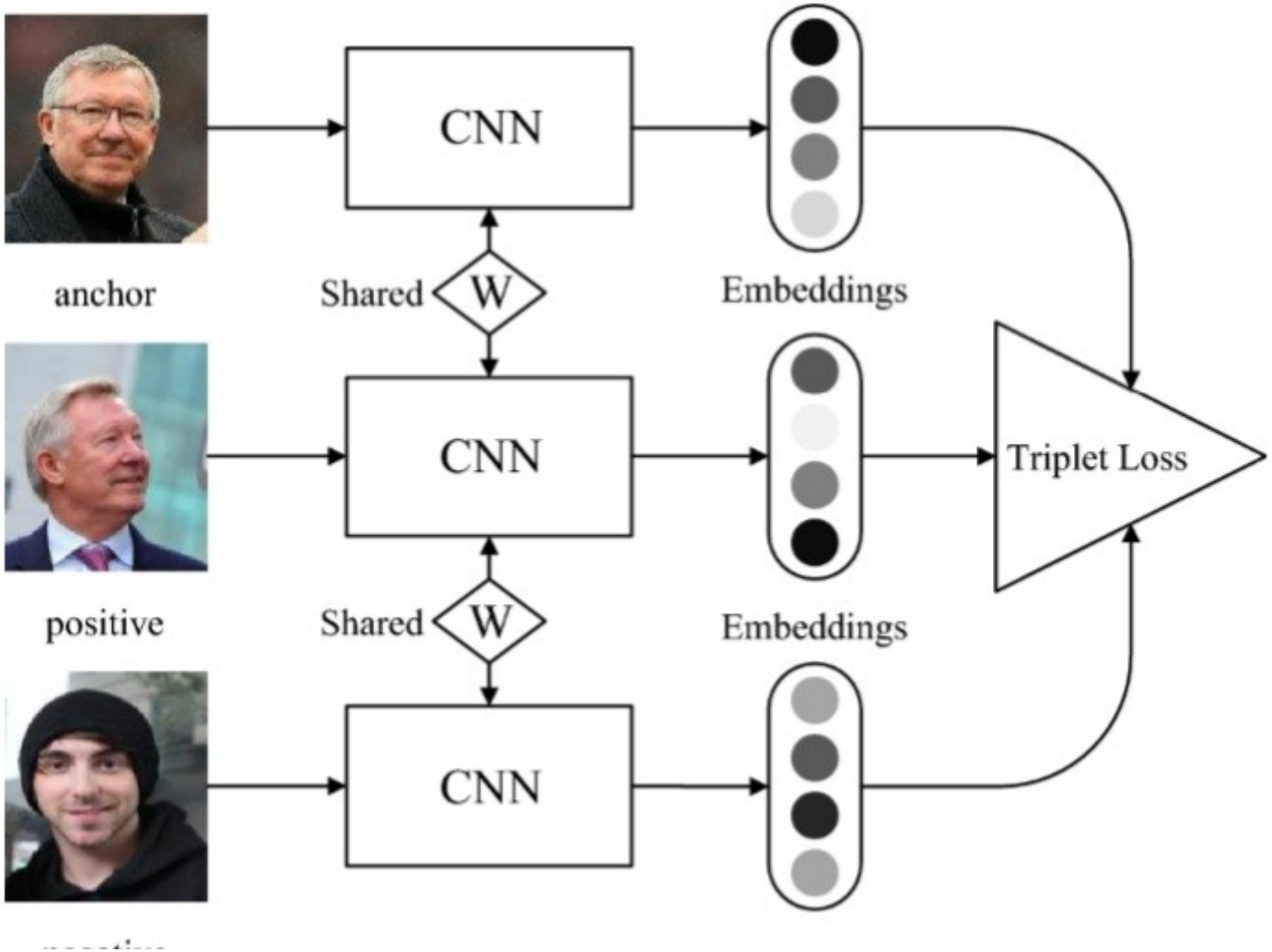


Siamese network used in Signet

$$L(s_1, s_2, y) = \alpha(1-y)D_w^2 + \beta y \max(0, m - D_w)^2$$

Y = 0 aynı
Y = 1 farklı
Dw öklid uzaklığı
M margin

Y=0 yakınlaşma
Y=1 uzaklaşma

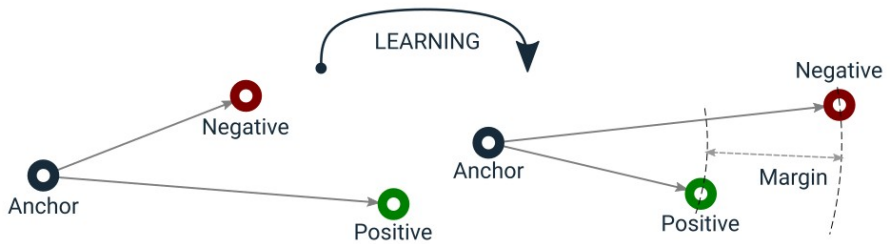

CNN
CNN
CNN
Shared W
Shared W
anchor
positive
Embeddings
Embeddings
Triplet Loss

Üçlü Görüntüler

Anchor Positive Negative

Triplet Loss Function

$$L(a, p, n) = \max\{d(a_i, p_i) - d(a_i, n_i) + \text{margin}, 0\}$$

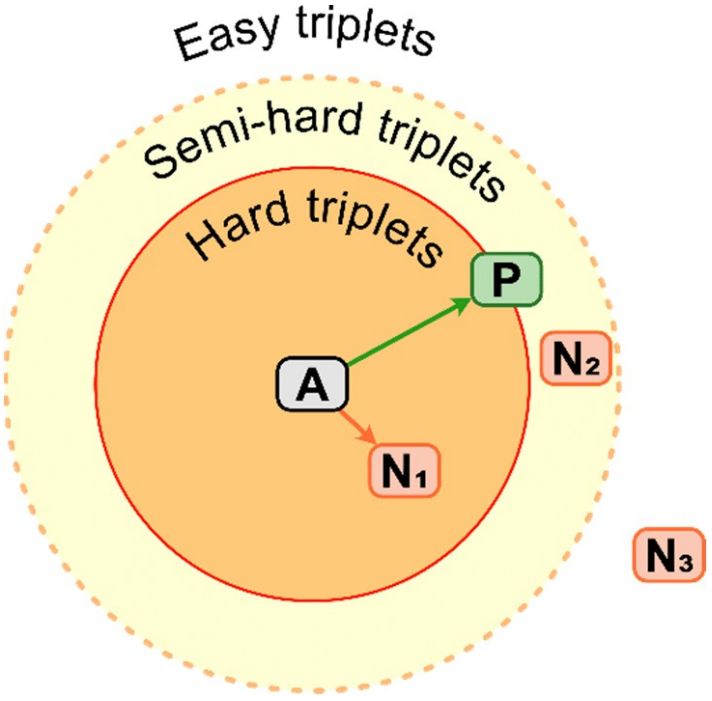


$$d(x_i, y_i) = \|\mathbf{x}_i - \mathbf{y}_i\|_p$$

- A- anchor N-negatif P-pozitif.
- Margin

Triplet’lerin Seçimi A,P,N

Anchor ve negatif görsel mümkün olduğunca yakın, anchor ve pozitif görsel birbirine uzak olmalıdır.



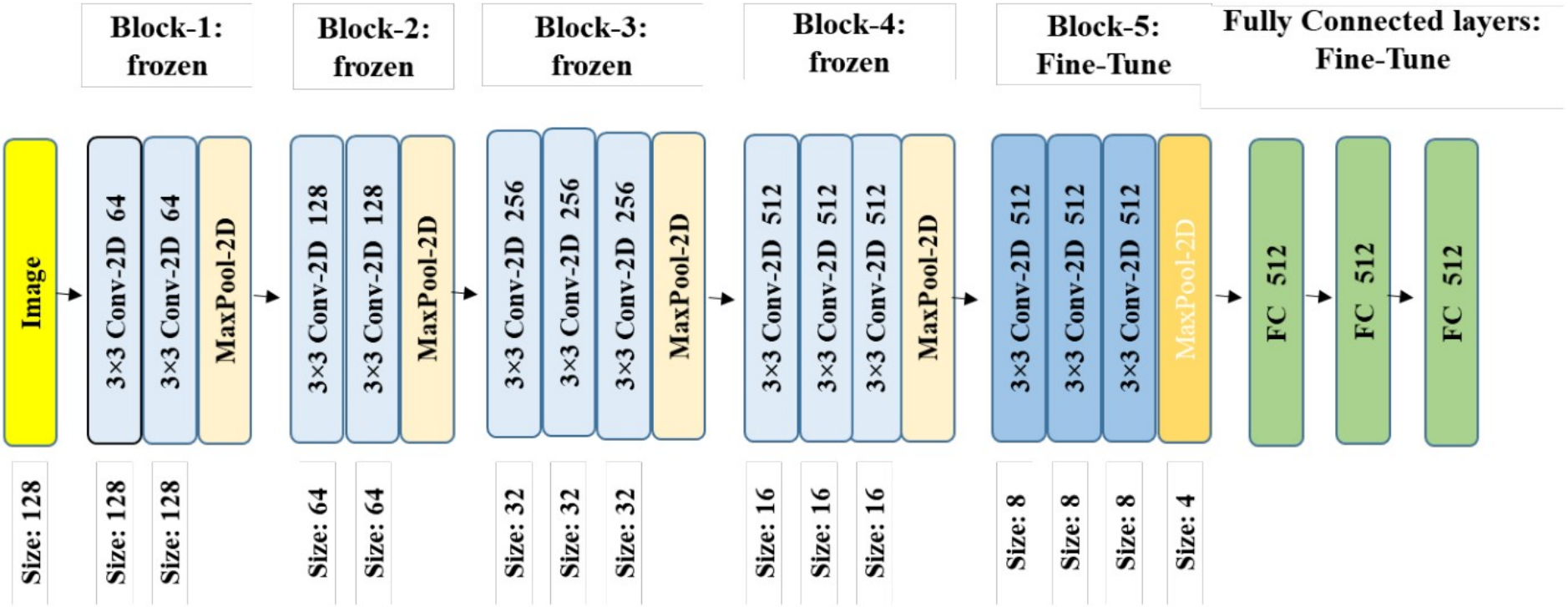
Transfer Öğrenme

- Bir problemi çözerken öğrenilmiş bilgiyi başka benzer bir problemin çözümünde bir kısmını yada tamamını kullanmaktır.
- Araba görsellerini tanımak için eğitilmiş bir ağın bir kısmı yada tamamı kamyon görsellerini tanımak için kullanılabilir.

CNN için Transfer Öğrenme

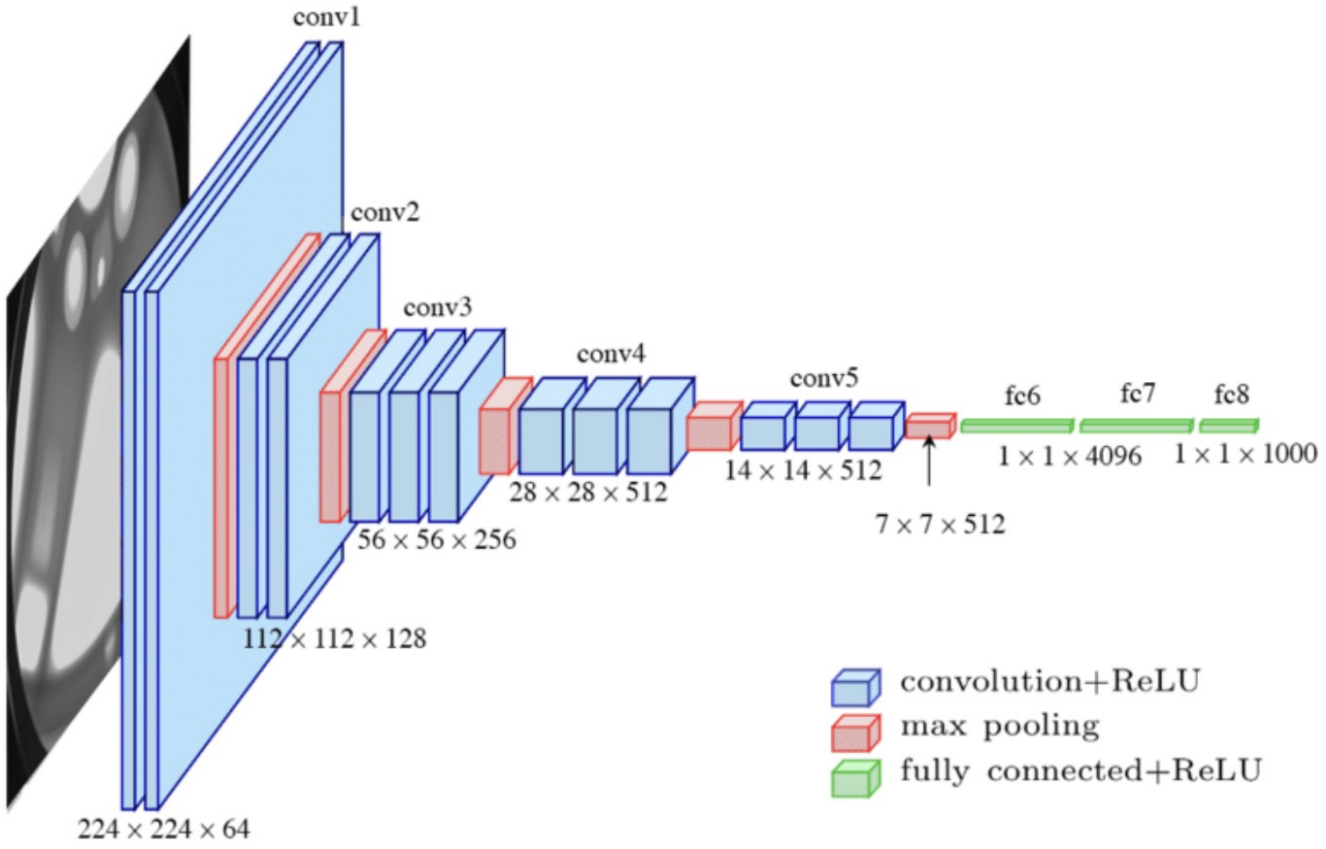
- Önceden eğitilmiş modelin ağırlıkları yüklenir.
- İlk katmanların ağırlıkları dondurulur.
- Yeni katmanlar eklenir.
- Yeni katmanlar eklenerek oluşturulmuş ağ eğitilir.
- Sonuca göre hiper-parametrelere ince ayar yapılabilir, yeni katmanlar eklenebilir.

Kullanılan Ağ



- M. Heidari and K. Fouladi-Ghaleh, "Using Siamese Networks with Transfer Learning for Face Recognition on Small-Samples Datasets," 2020 International Conference on Machine Vision and Image Processing (MVIP), 2020, pp. 1-4, doi: 10.1109/MVIP49855.2020.9116915.

VGG-16

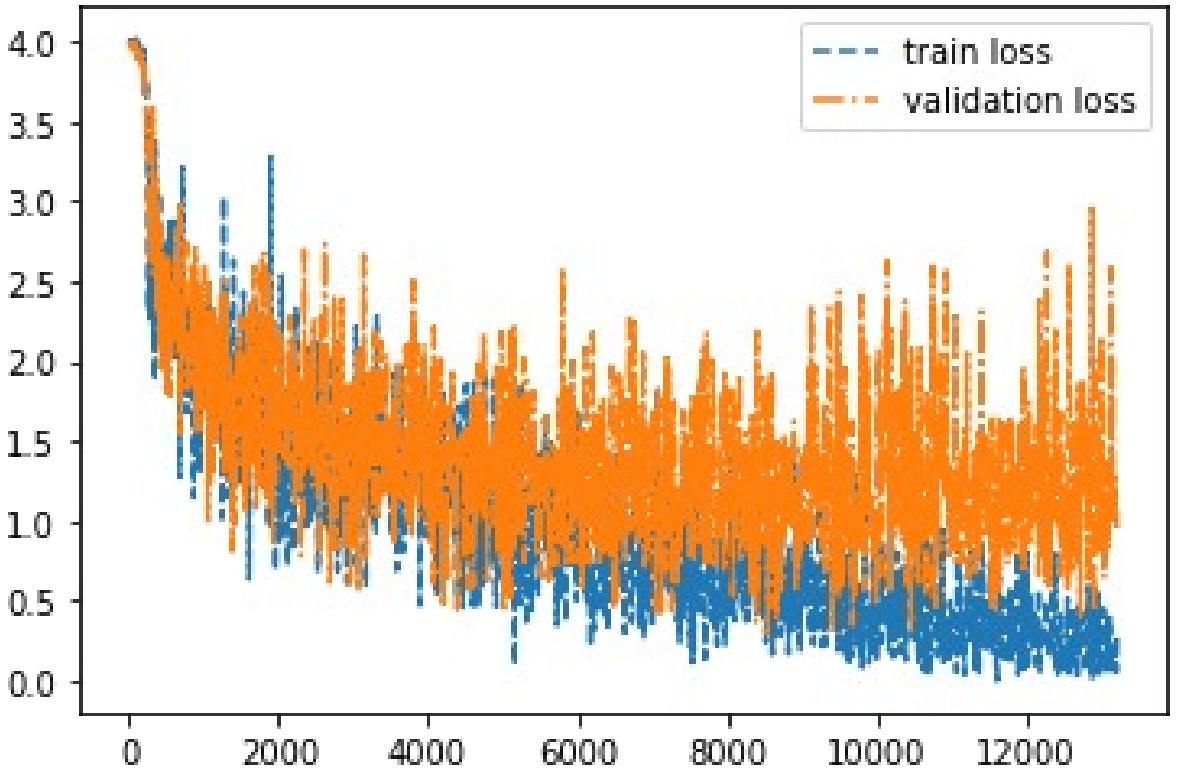

conv1
conv2
conv3
conv4
conv5
fc6
fc7
fc8
224 × 224 × 64
112 × 112 × 128
56 × 56 × 256
28 × 28 × 512
14 × 14 × 512
7 × 7 × 512
1 × 1 × 4096
1 × 1 × 1000
convolution+ReLU
max pooling
fully connected+ReLU

Eğitilen Ortam

- PyTorch ile geliştirildi.
- Google Colab kullanılarak eğitildi.
- Casia Web Faces
- Labeled Faces in the Wild (LFW)
- FERET Colored

Ağın Eğitilmesi

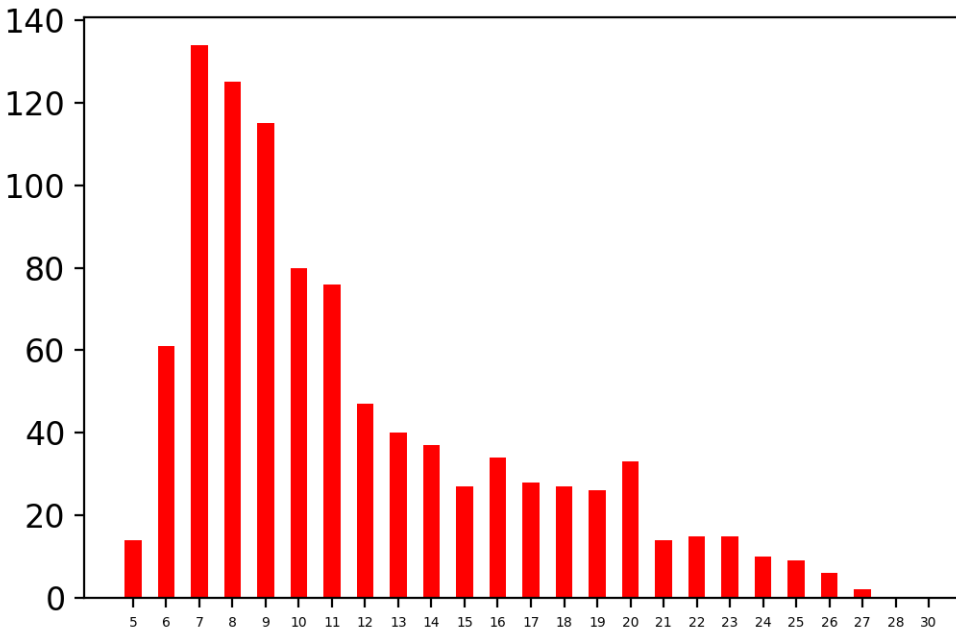
- Casia Web Faces
- 494414 Yüz içeren içinden rastgele seçilmiş 268080 görüntü kullanıldı.
- 89360 Adet ikisi aynı kişiye ait biri ise farklı triplet'ler oluşturuldu ve eğiltildi.
- Batch size 40 olarak seçildi. Oluşan 2234 Batch %20 - % 80 olarak eğitim ve doğrulama seti olarak bölündü

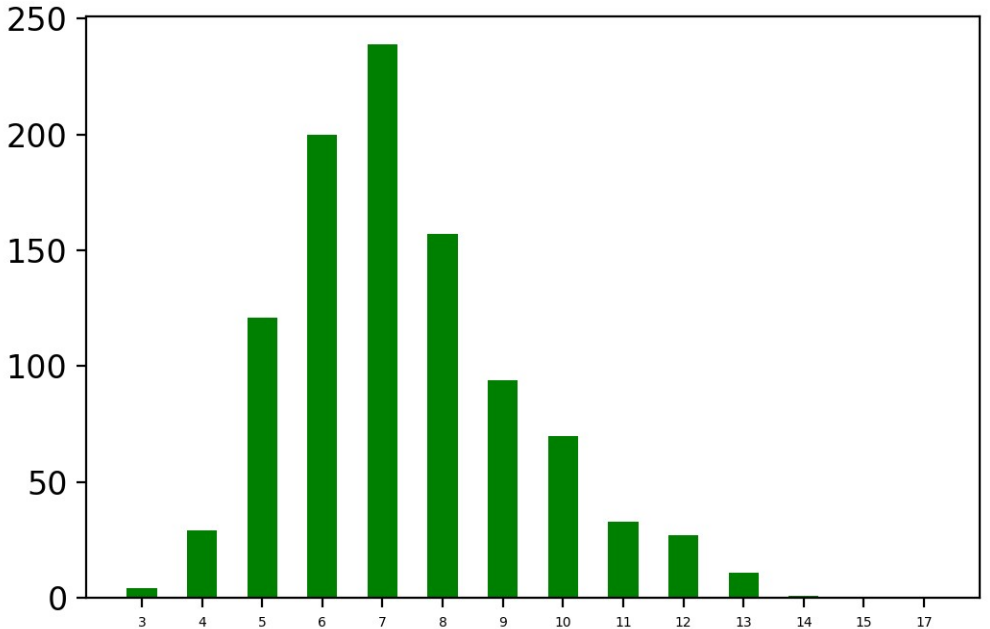

train loss
validation loss
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
2000
4000
6000
8000
10000
12000

Eğitim Süresine Göre Sonuçlar

- Casia Web Faces vertabanından rastgelen seçilmiş görseller ile 1 – 11 epoch arası eğitildi.
- Her epoch sonunda eğitilen model kaydedildi.
- Kaydedilen her model LFW veritabanıdaki 13233 görselden oluşturulan üçlü setler ile test edildi.

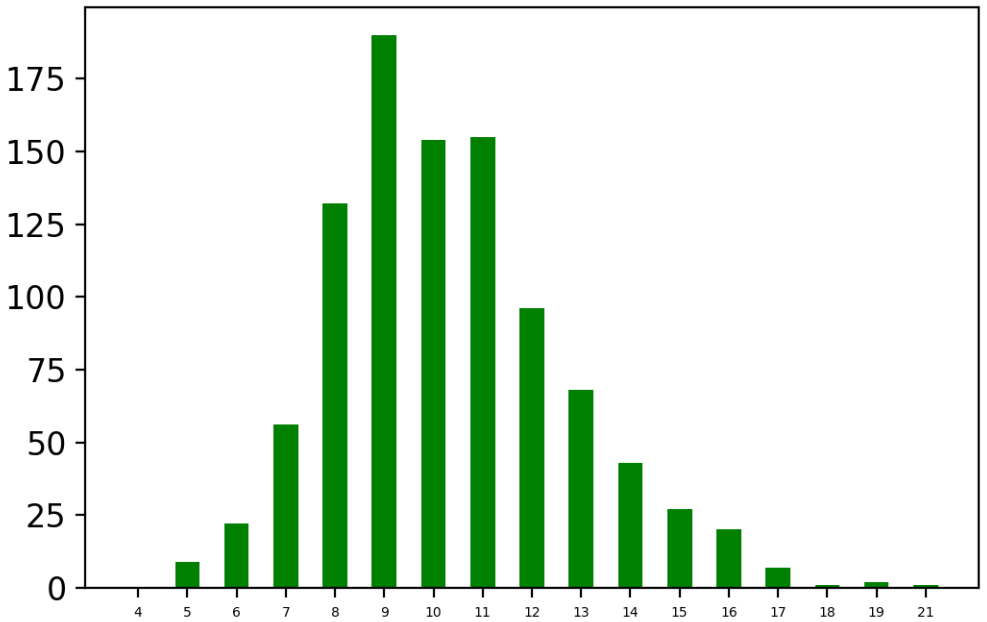
1 Epoch

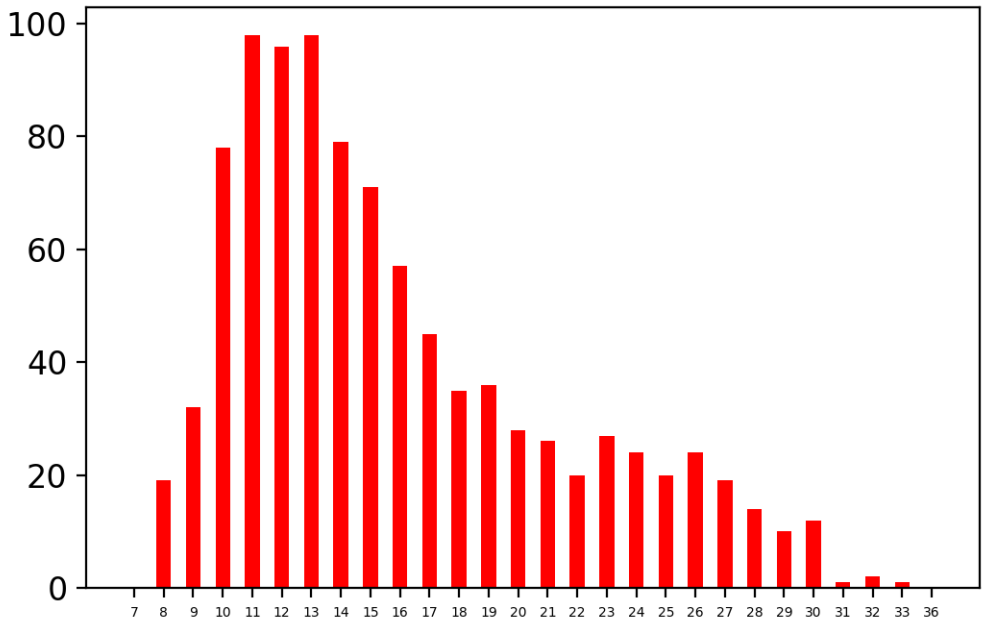





```
Eşik değeri – doğruluk – tp, tn, fn, fp
8.589218473434432 0.7095 (766, 653, 234, 347)
```

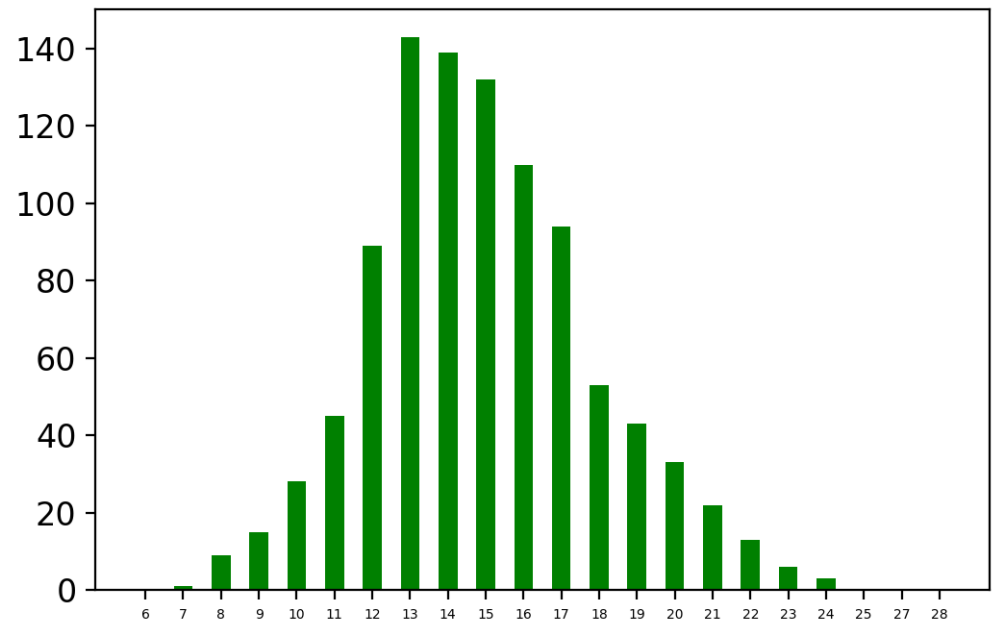
2 Epoch

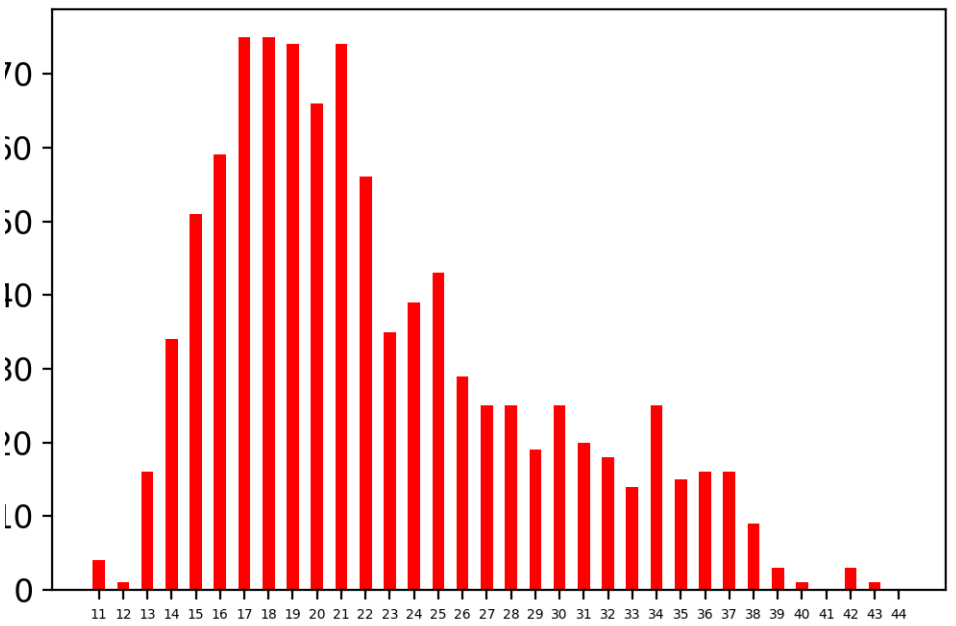





```
Eşik değeri – doğruluk – tp, tn, fn, fp
11.94092807769773 0.7525 (776, 729, 224, 271)
```

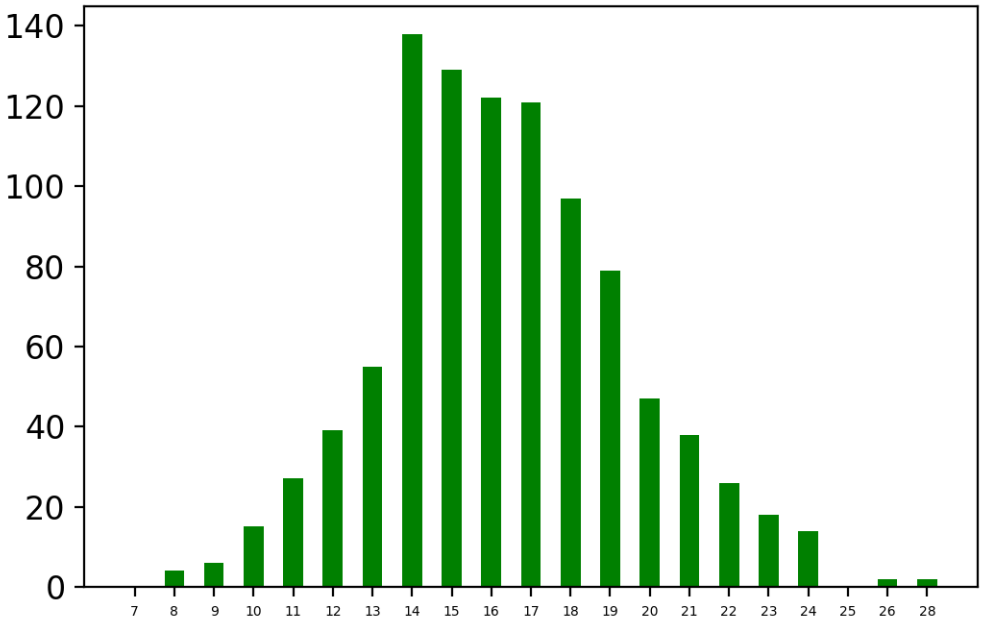
4 Epoch

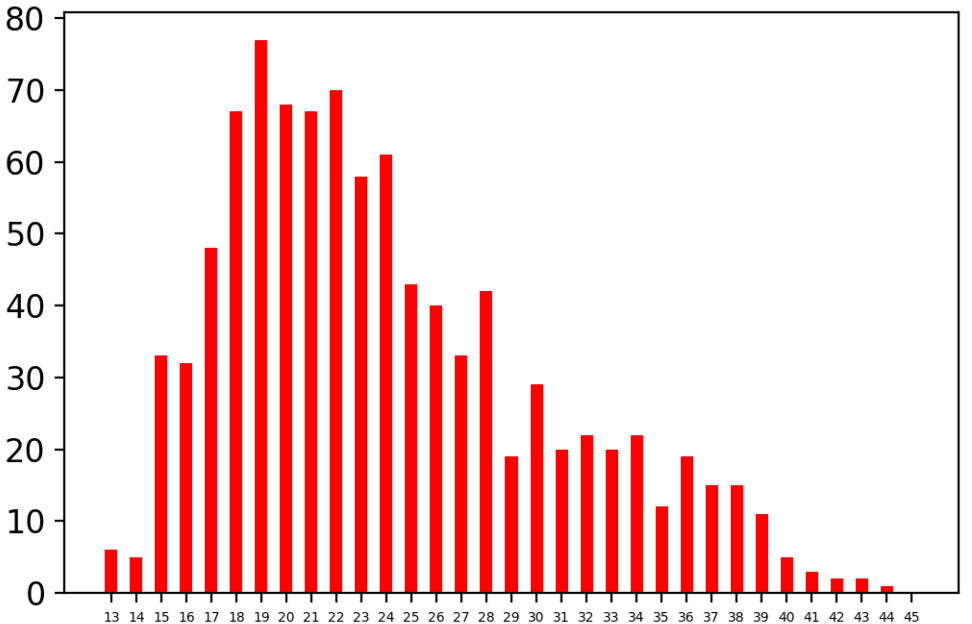





```
Eşik değeri – doğruluk – tp, tn, fn, fp
17.076625919342007 0.7885 (788, 789, 212, 211)
```

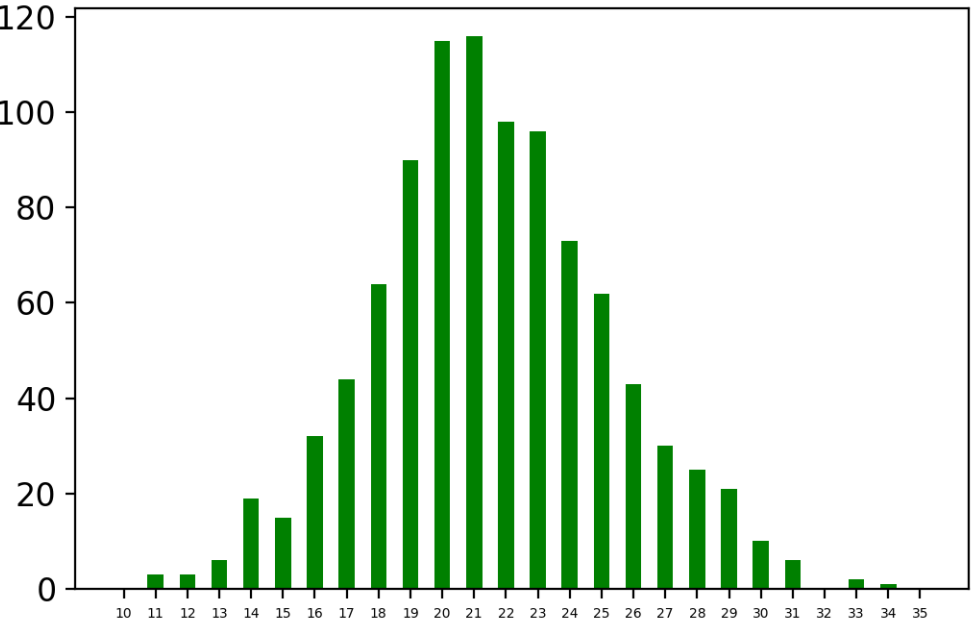
5 Epoch

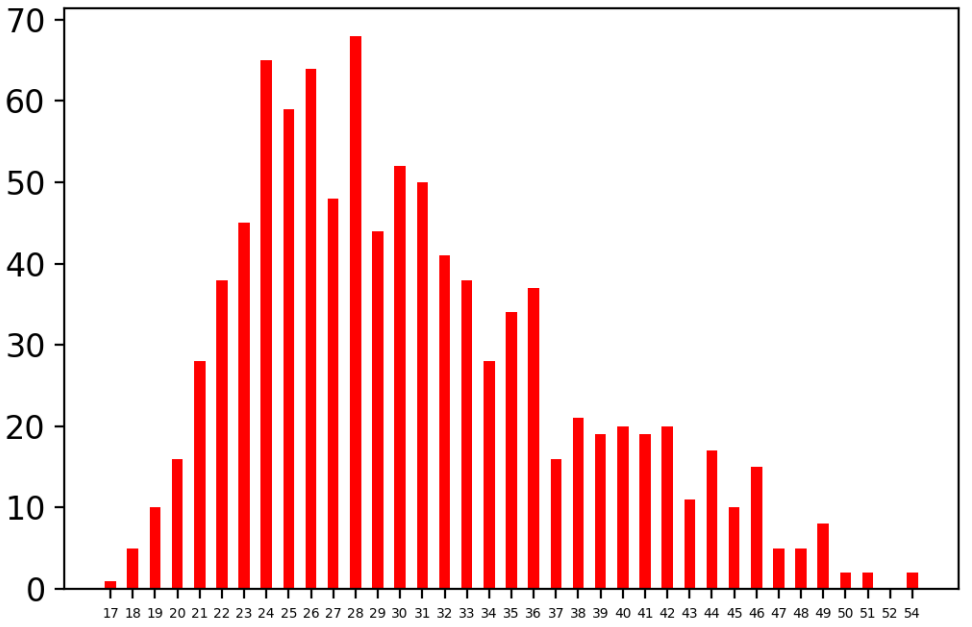





```
Eşik değeri – doğruluk – tp, tn, fn, fp
18.928353595733604 0.792 (804, 780, 196, 220)
```

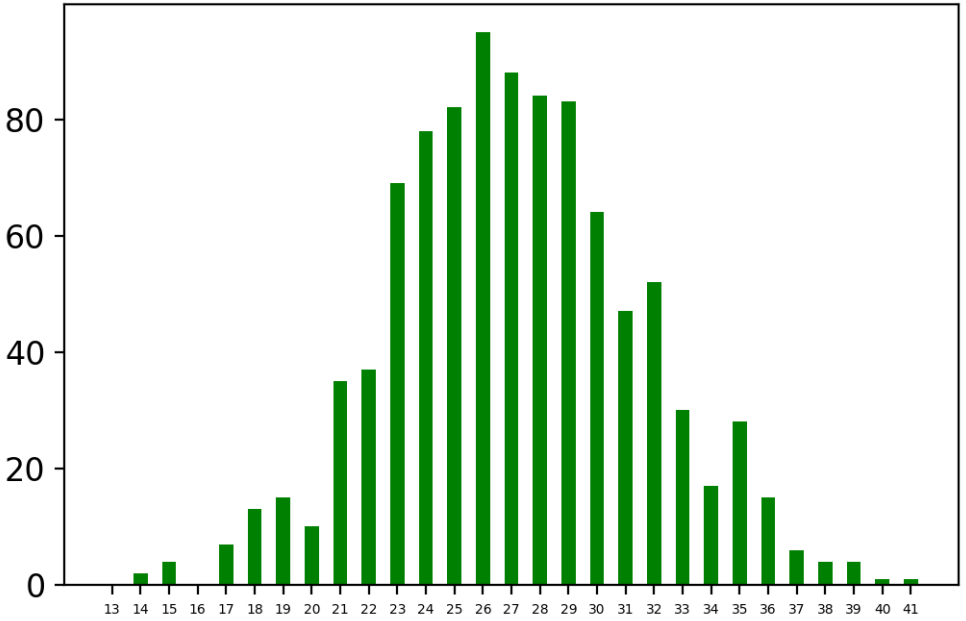
7 Epoch

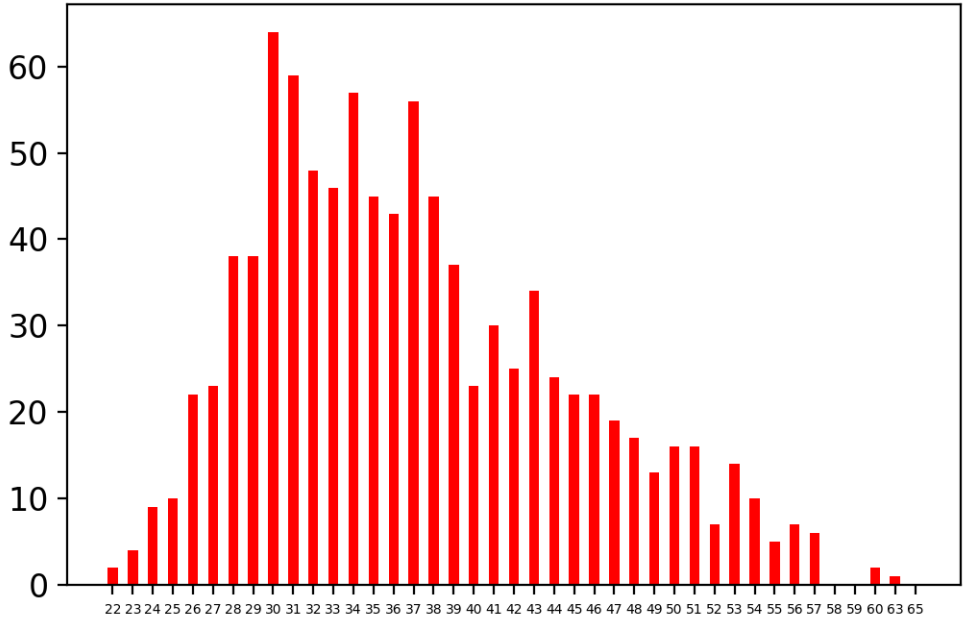





```
23.98485965728755 0.792 (758, 826, 242, 174)
```

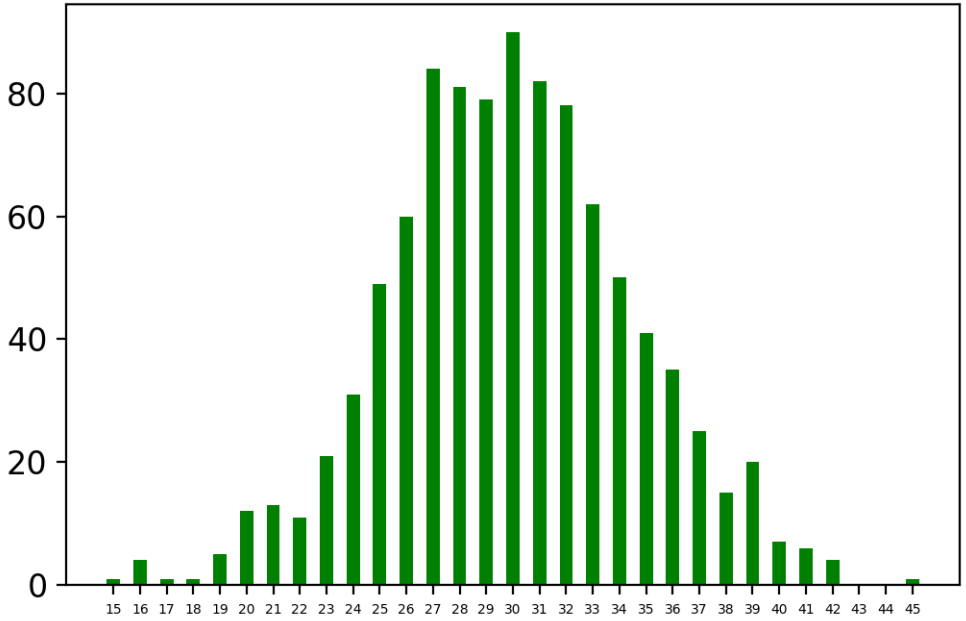
9 Epoch

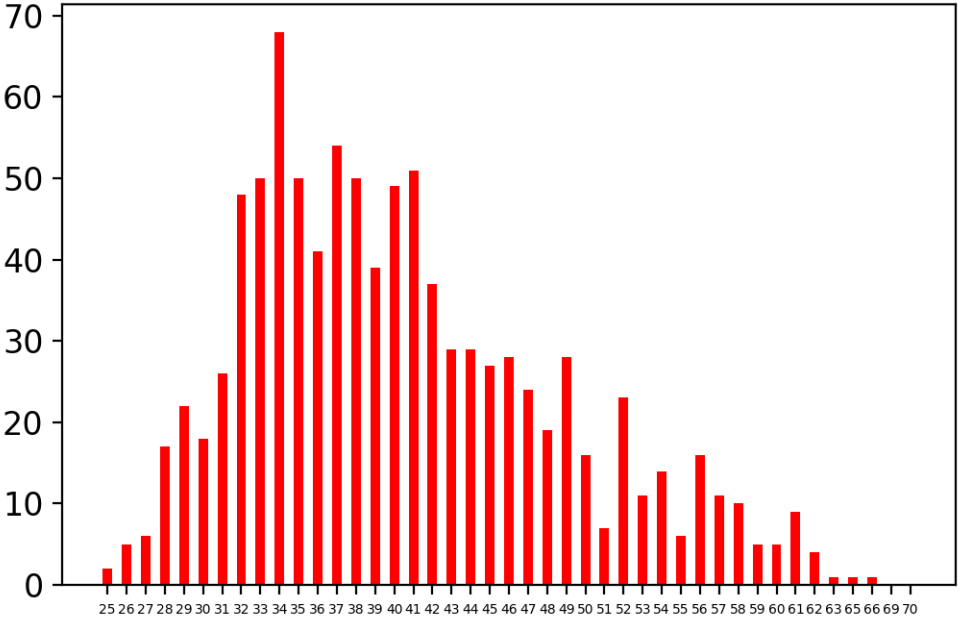





```
29.83460254669184 0.7875 (743, 832, 257, 168)
```

10 Epoch






```
33.38655490875238 0.7905 (780, 801, 220, 199)
```

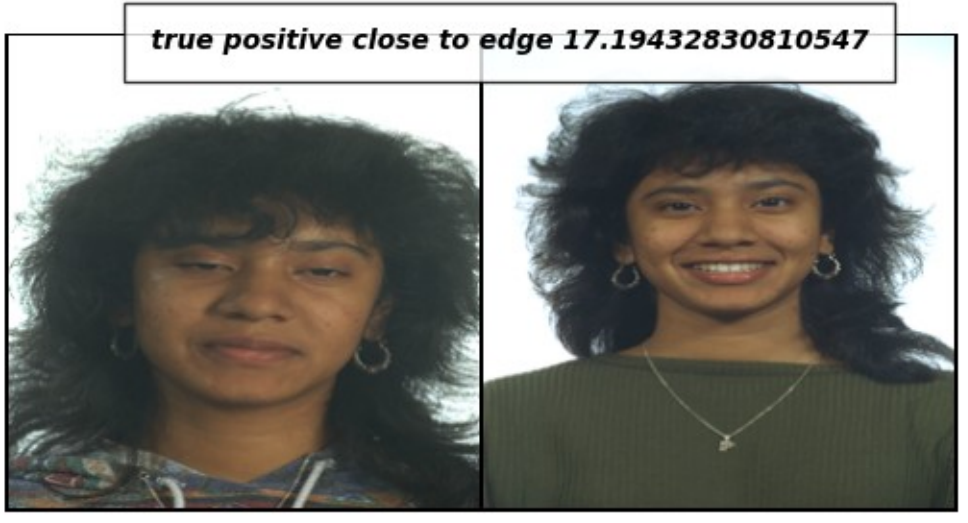
Feret Color Faces Database

- Büyük bir kısmı kontrollü ortamda çekilmiş görüntülerden oluşmakta.
- Sadece ön yüz görselleri ile test edildi.
- 725 kişiye ait 1695 adet yüz görselinden rastgele 1130 çift oluşturuldu.
- %89 doğruluğa ulaşıldı.


true positive good match 7.60389518737793


true positive good match 7.971672058105469


true positive good match 6.899694442749023


true positive close to edge 17.19432830810547

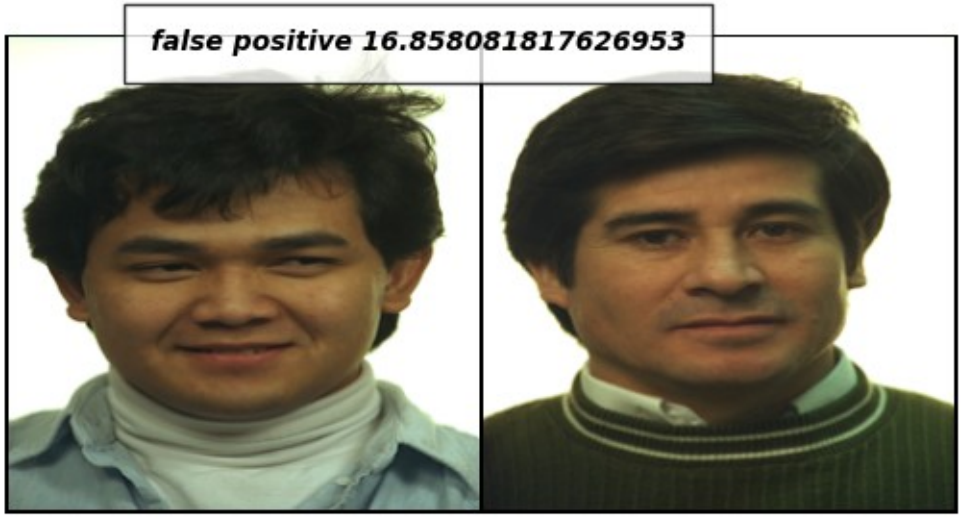

false negative 19.273466110229492


false negative 20.282981872558594


false negative 27.918893814086914


false negative 21.303600311279297


false positive 10.948031425476074


false positive 16.858081817626953


false positive 16.51185417175293


false positive 11.388476371765137

Daha iyi üçlü görsellerin oluşturulması.

- İyi seçilmemiş üçlü görsellerden ağ için çok öğrenecek bir şey yok.
- Veri setinin bir kısmı ayrıldı. Ayrılan kısım ile eğitim sırasında ağın güncel haliyle “zor” olarak sınıflandırılan üçlü görseller seçildi ve eğitim setine dahil edildi.
- Vakit alan bir süreç.

Daha iyi üçlü görsellerle eğitilen modelin değelendirilmesi.

- Casia verisetinden rastgele seçilerek oluşuturulan üçlülerle eğitilen modelin ağırlıkları kullanıldı.
- LFW setiyle anlatılan metot kullanıldı.
- Daha iyi sonuç alındı. Aynı görseller ile tüm üçlüler için %92.6 doğruluğa erişildi.


im 3.9840593338012695


im 3.944350242614746


false positive 5.803426265716553


false positive 6.970326900482178

Daha iyi üçlü görsellerle eğitilen modelin değelendirilmesi.

- Daha “zor” üçlü görseller ile eğitildiğinde daha iyi sonuçlar elde edildi.
- Daha kaliteli veriyle ve farklı metotlarla üçlü görseller elde edilerek daha iyi sonuçlar elde edilebilir.

Demo

- Eğtilen yüz tanıma, tespiti sınıflarının etrafına yeni sınıflar yazıldı.
- QT kullanılarak basit bir arayüz oluşturuldu.

Kaynakça

- M. Heidari and K. Fouladi-Ghaleh, "Using Siamese Networks with Transfer Learning for Face Recognition on Small-Samples Datasets," 2020 International Conference on Machine Vision and Image Processing (MVIP), 2020, pp. 1-4, doi: 10.1109/MVIP49855.2020.9116915.
- F. Schroff, D. Kalenichenko and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 815-823, doi: 10.1109/CVPR.2015.7298682.
- K. Vikram and S. Padmavathi, "Facial parts detection using Viola Jones algorithm," 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), 2017, pp. 1-4, doi: 10.1109/ICACCS.2017.8014636.
- Hastie, T.; Tibshirani, R. & Friedman, J. (2001), The Elements of Statistical Learning , Springer New York Inc. , New York, NY, USA .
- https://www.deeplearning.ai/
- StatQuest with Josh Starmer

Dinlediğiniz İçin Teşekkürler.