Boyutlandırma

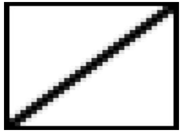
Resizing

Dört yöntem

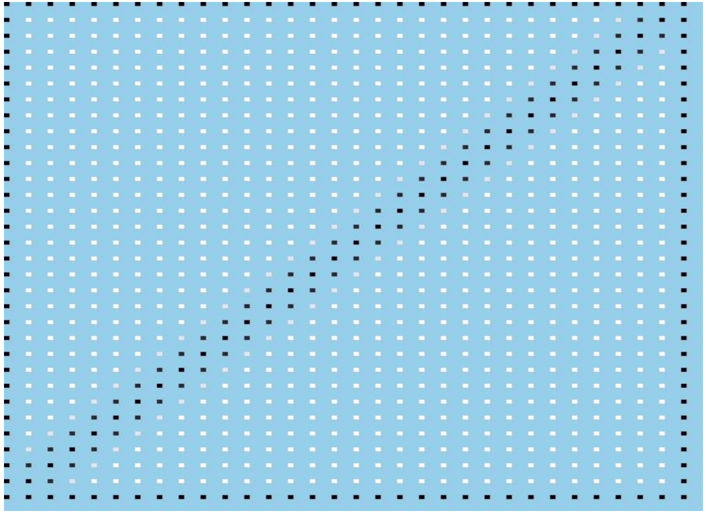
- Piksel çoğullama/silme
- En yakın komşu
- Bilinear interpolasyon
- Bicubic interpolasyon

Piksel çoğullama ile görüntü büyütme

Örnek
Bu görüntünün
4 kat büyük
hali

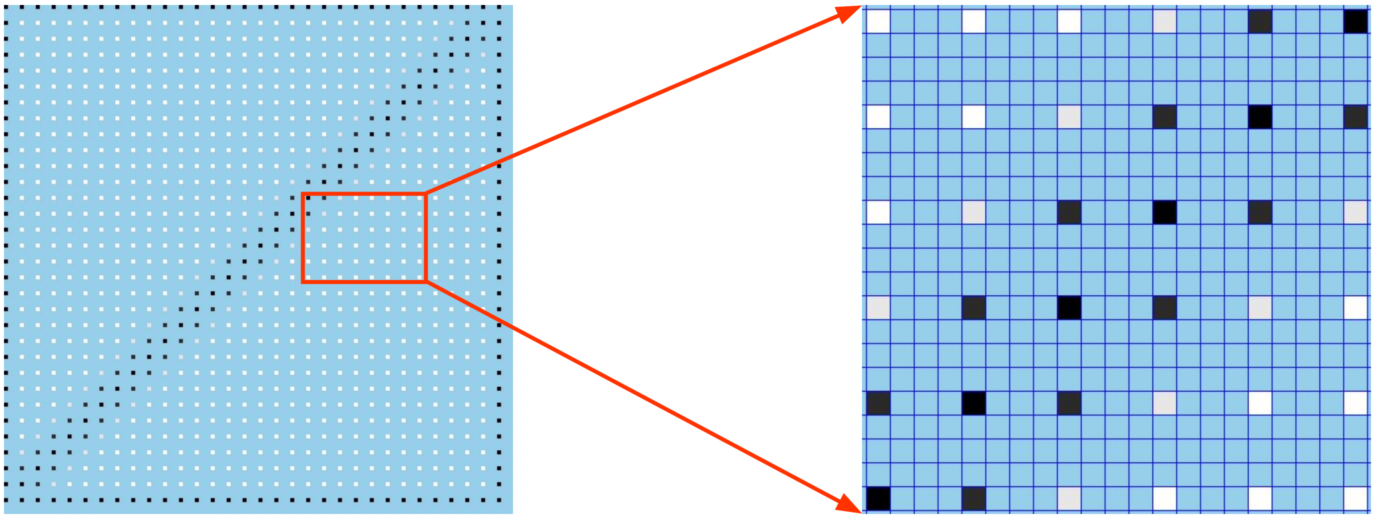


4 kat büyük boyutlu boş bir matrisle başlarız.



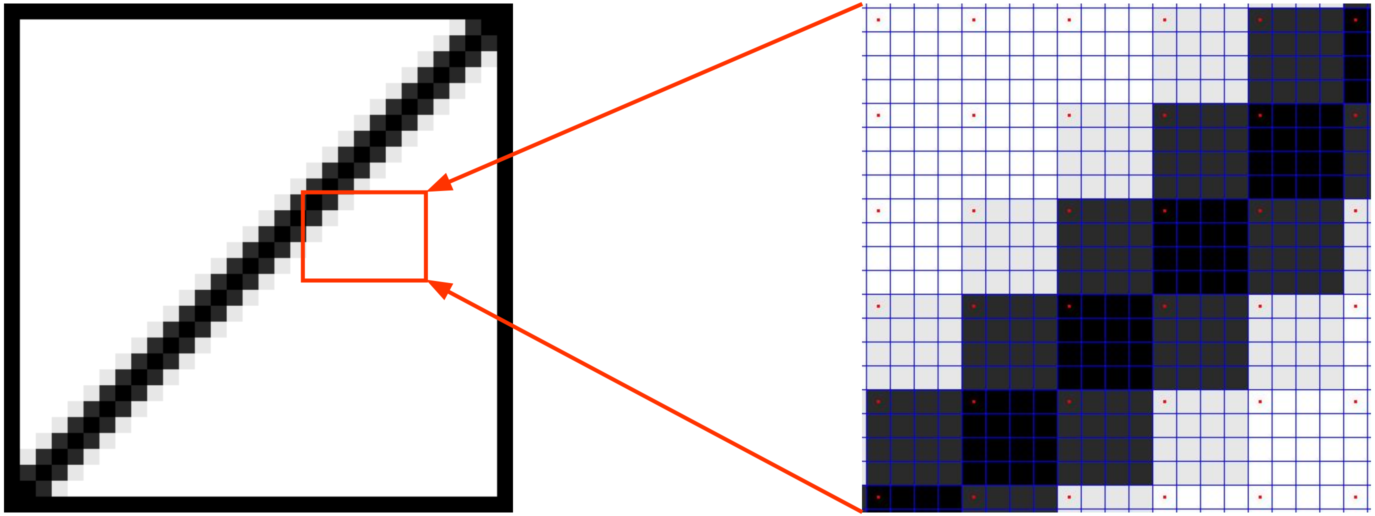
Orijinal görüntü değerlerini yeni matrisin her 4. satırın 4. sonraki elemanına gelecek şekilde yerleştir.

Piksel çoğullama ile görüntü büyütme



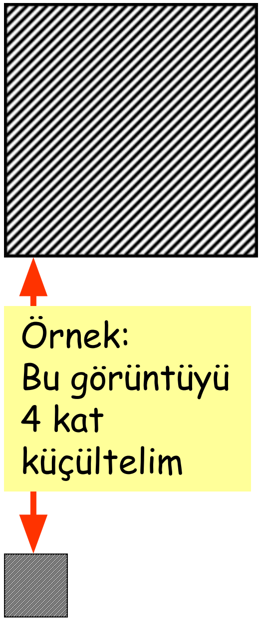
Detaylı gösterim

Piksel çoğullama ile görüntü büyütme



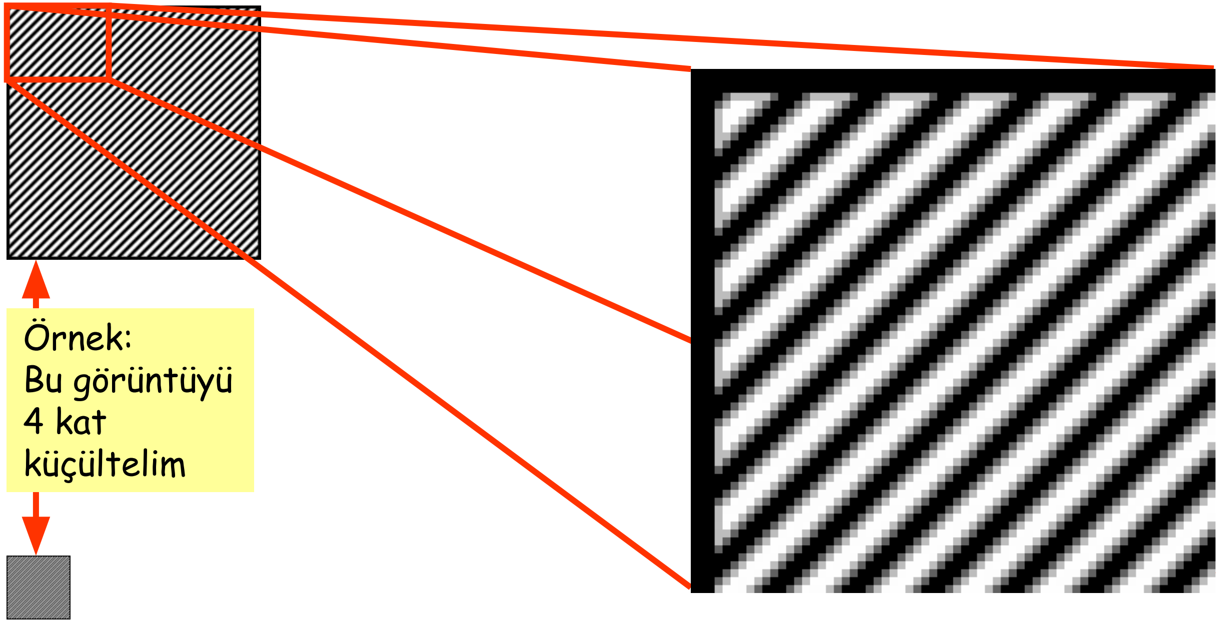
Orijinal piksel değerini 4x4 lük alana çoğullarız

Piksel silme ile görüntü küçültme

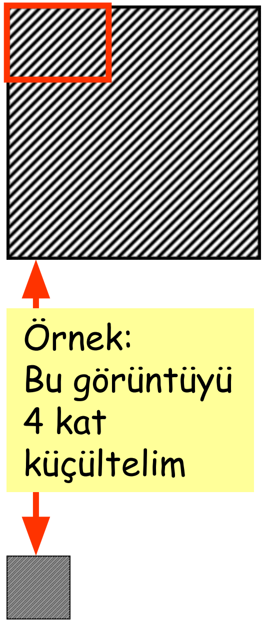


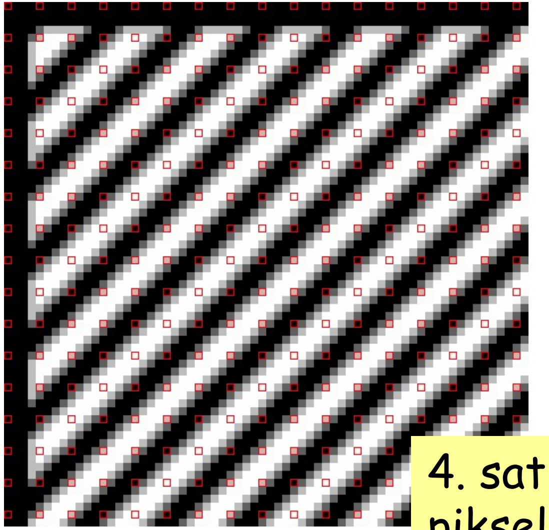
Her satırdaki n. piksel değerini tut.

Piksel silme ile görüntü küçültme

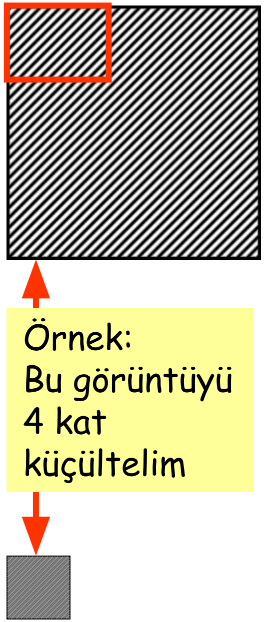


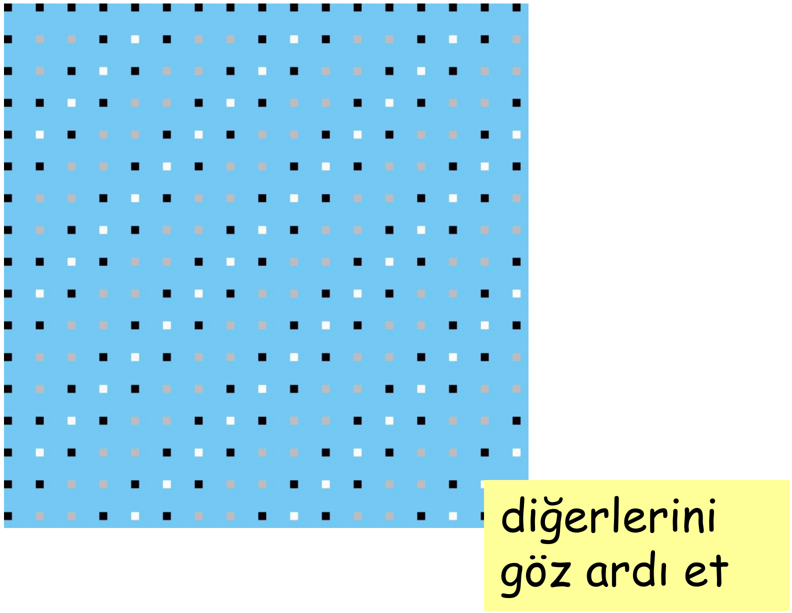
Piksel silme ile görüntü küçültme



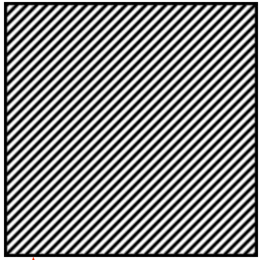


Piksel silme ile görüntü küçültme

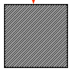


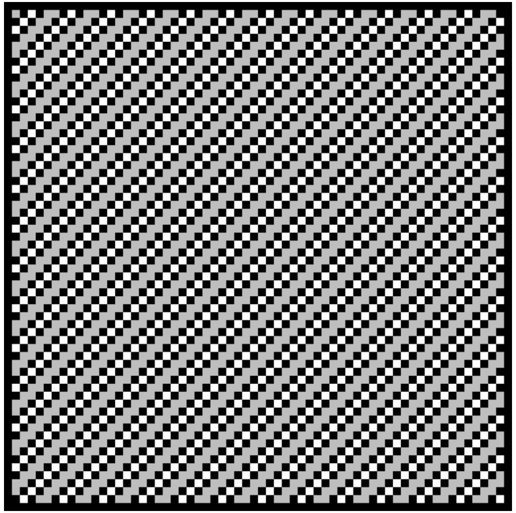


Piksel silme ile görüntü küçültme



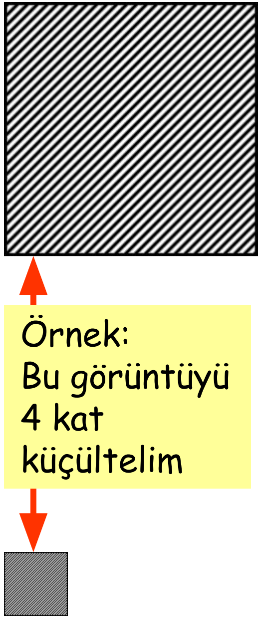
Örnek:
Bu görüntüyü
4 kat
küçültelim

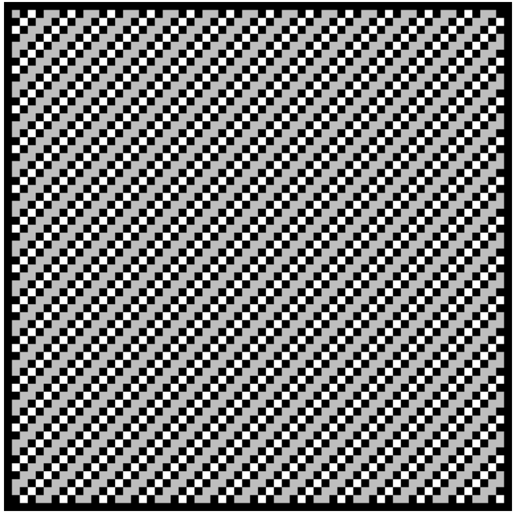




Kalanları
yeni bir
görüntüye
kopyala

Piksel silme ile görüntü küçültme

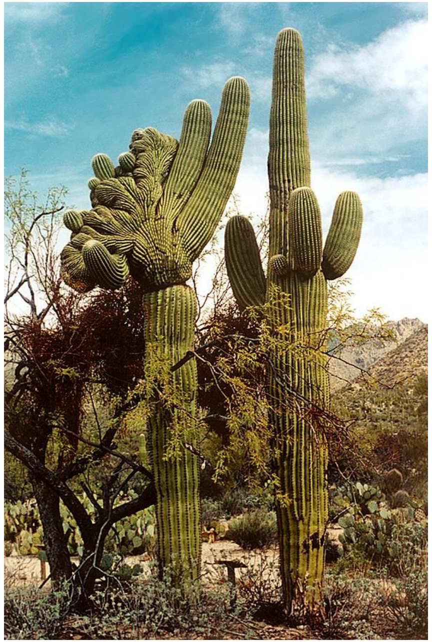




En yakın komşu boyutlama

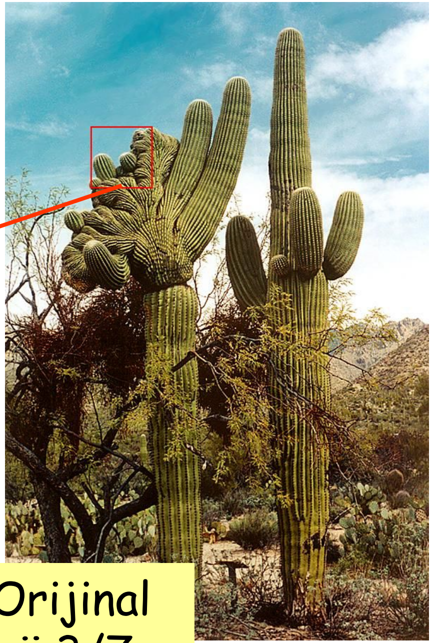
"En yakın komşu" algoritması piksel çoğullama ve silme yöntemlerinin genelleştirilmiş halidir.

Aynı zamanda kesirsel boyutlanmaya izin verir. Örneğin p/q oranında boyutlandırma yapılabilir.



En yakın komşu boyutlama

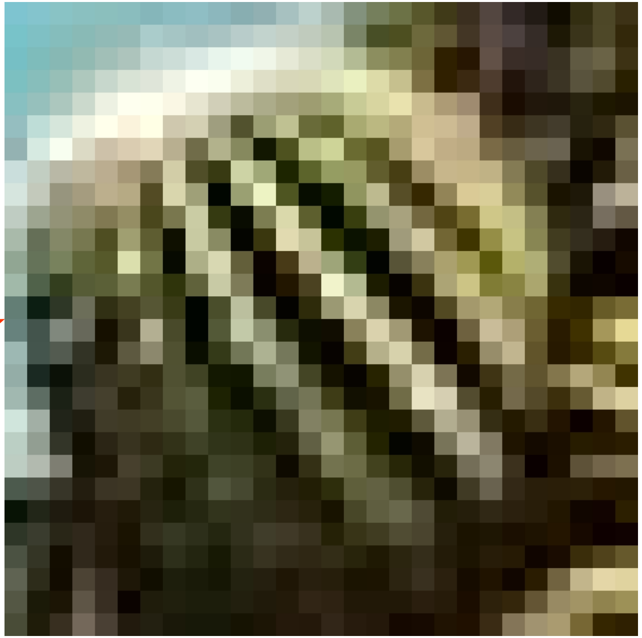
İşlem sonucunu daha iyi anlamak için yakından bakalım



Örnek: Orijinal görüntüyü 3/7 oranında boyutlandır.

En yakın komşu boyutlama

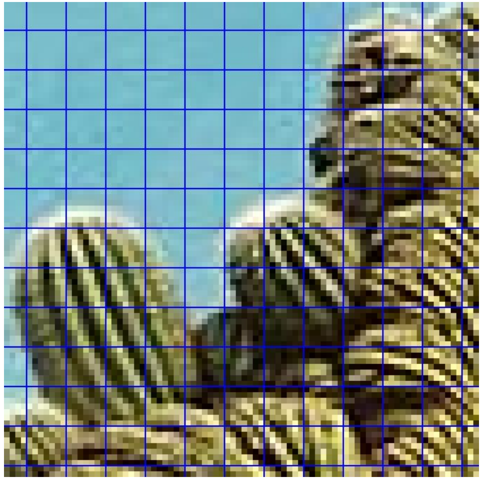
3/7 boyutlandır

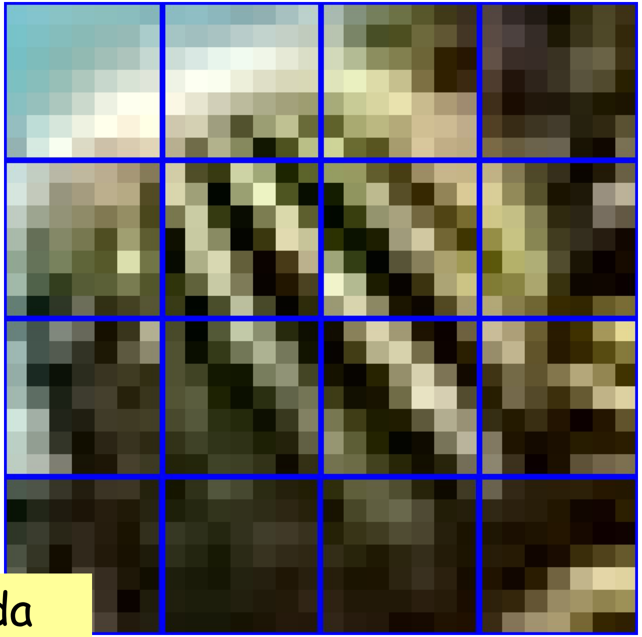


Zoom

En yakın komşu boyutlama

3/7 boyutlandır

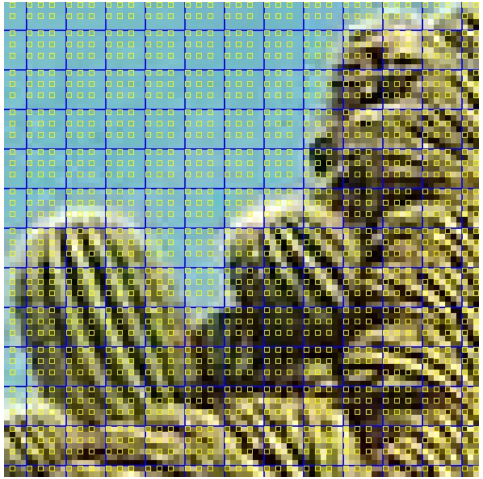


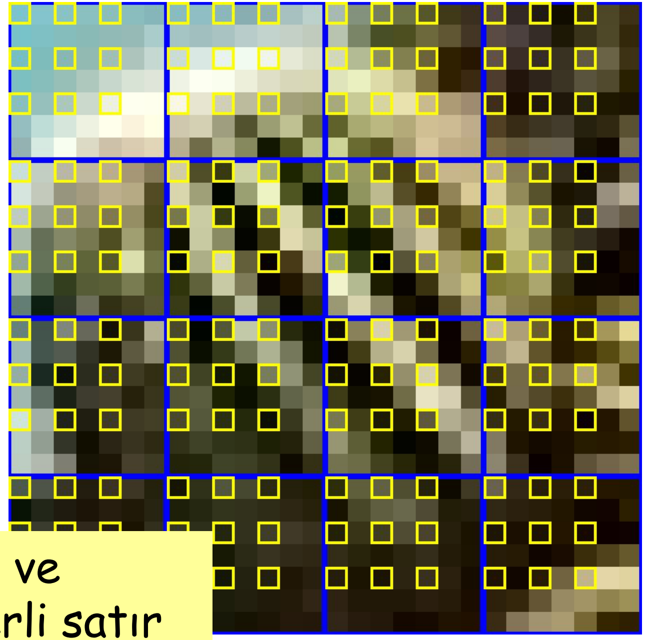


7x7 oranında gridle

En yakın komşu boyutlama

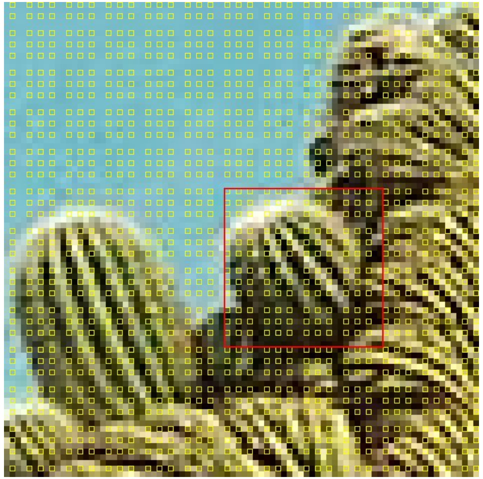
3/7 boyutlandır





Her 7 satır ve sütunda 3'erli satır ve sütunları seç (Sarı)

En yakın komşu boyutlama

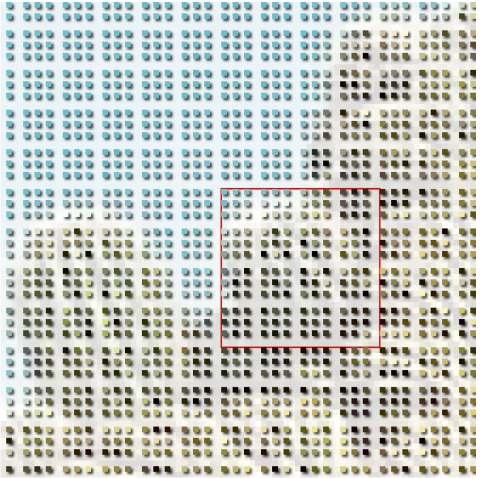


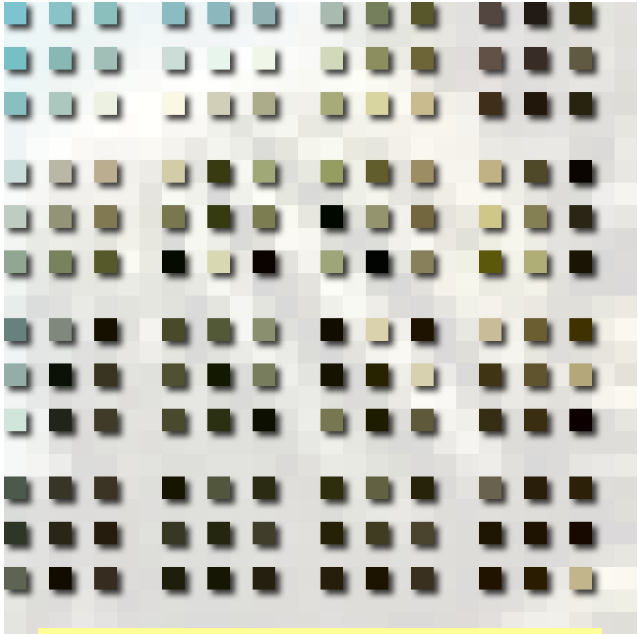
3/7 boyutlandır

Sarı pikselleri tut

En yakın komşu boyutlama

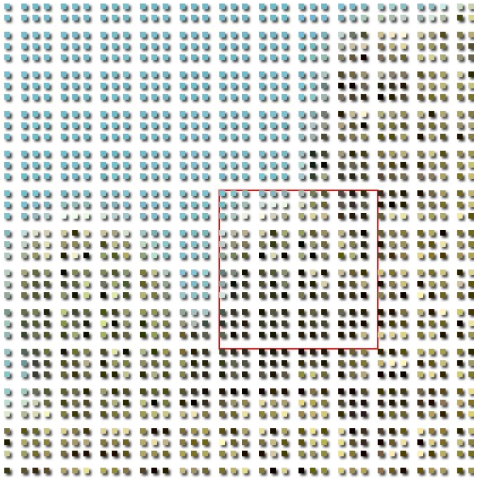
3/7 boyutlandır



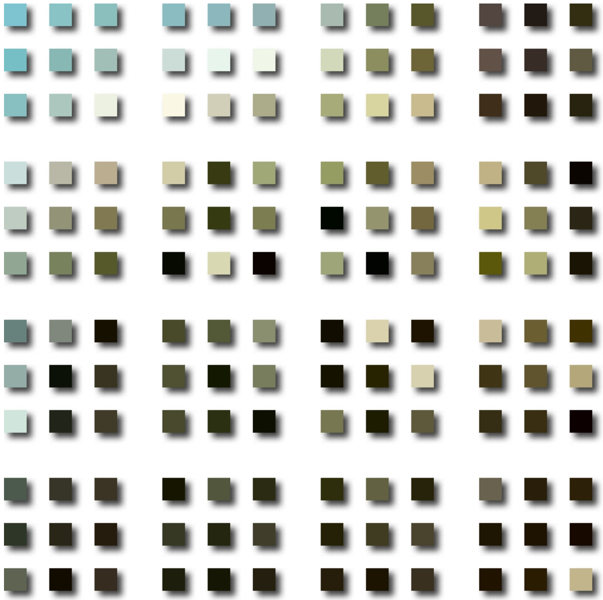


Diğer pikselleri tutma

En yakın komşu boyutlama



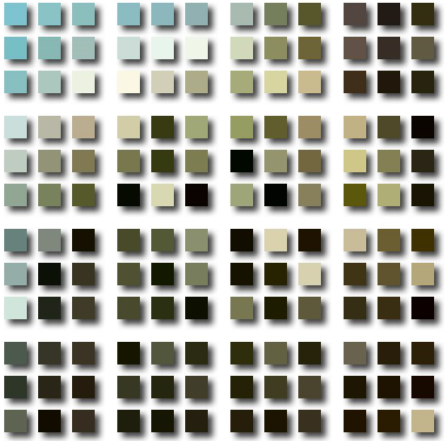
3/7 boyutlandır



Sarı pikselleri kopyala

En yakın komşu boyutlama

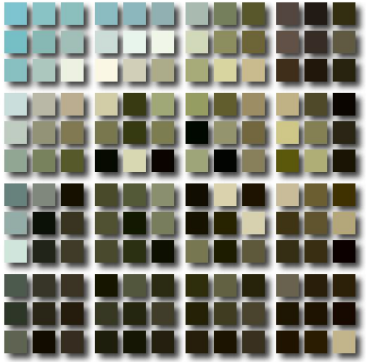
3/7 boyutlandır



yeni bir görüntüye aktar

En yakın komşu boyutlama

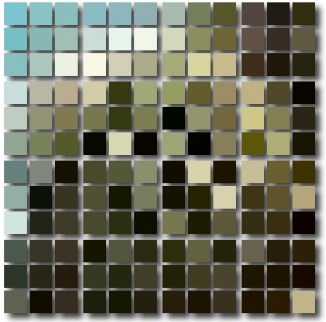
3/7 boyutlandır



yeni bir görüntüye aktar

En yakın komşu boytlama

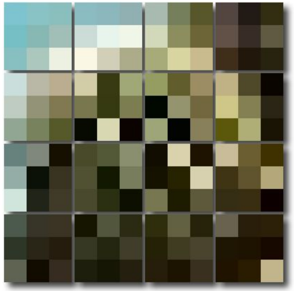
3/7 boyutlandır



yeni bir
görüntüye aktar

En yakın komşu boyutlama

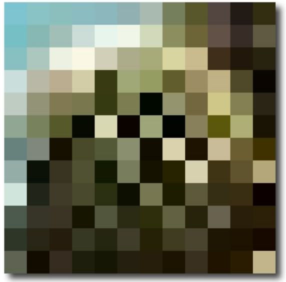
3/7 boyutlandır



yeni bir
görüntüye aktar

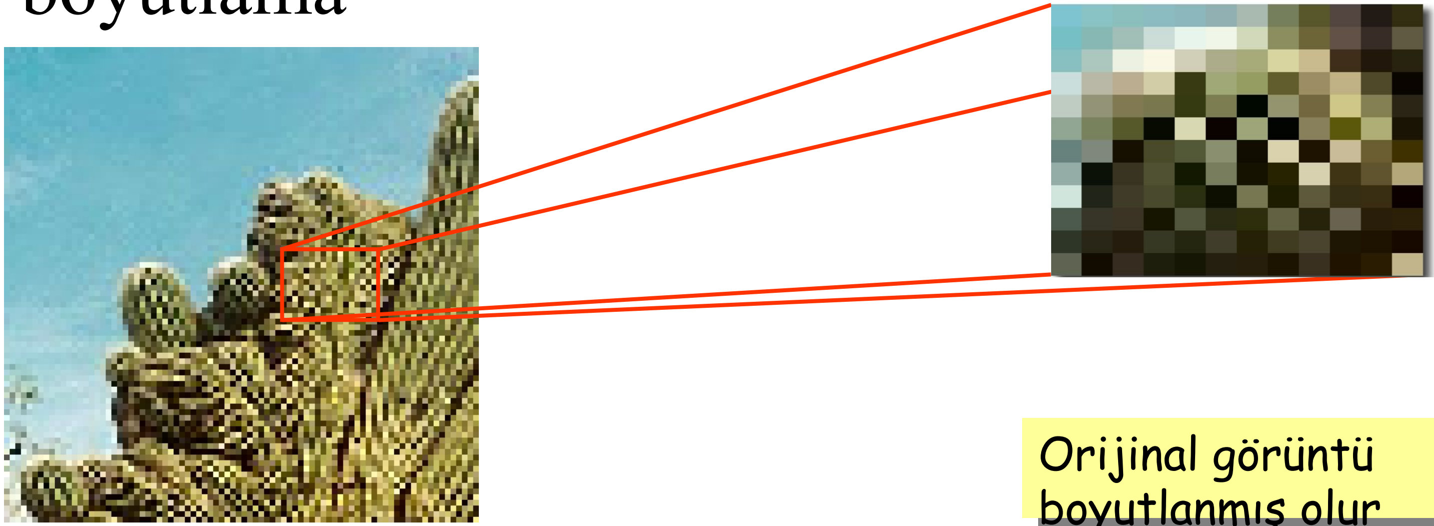
En yakın komşu boyutlama

3/7 boyutlandır



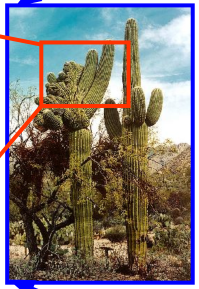
Orijinal görüntü boyutlanmış olur

En yakın komşu boyutlama



Orijinal görüntü boyutlanmış olur

En yakın komşu boyutlama



En yakın komşu boyutlama

Orijinal görüntü

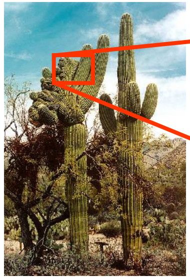
7/3 boyutlandır

Pikselleri 7/3 boyutu için dışa yay

Sonra araları doldur

En yakın komşu boyutlama

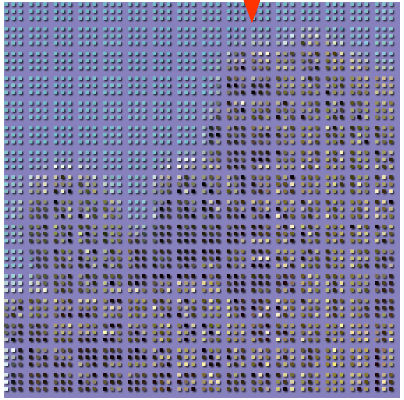
7/3 boyutlandır



Orijinal görüntü

Buradaki her bir 3x3 lük görüntüyü al

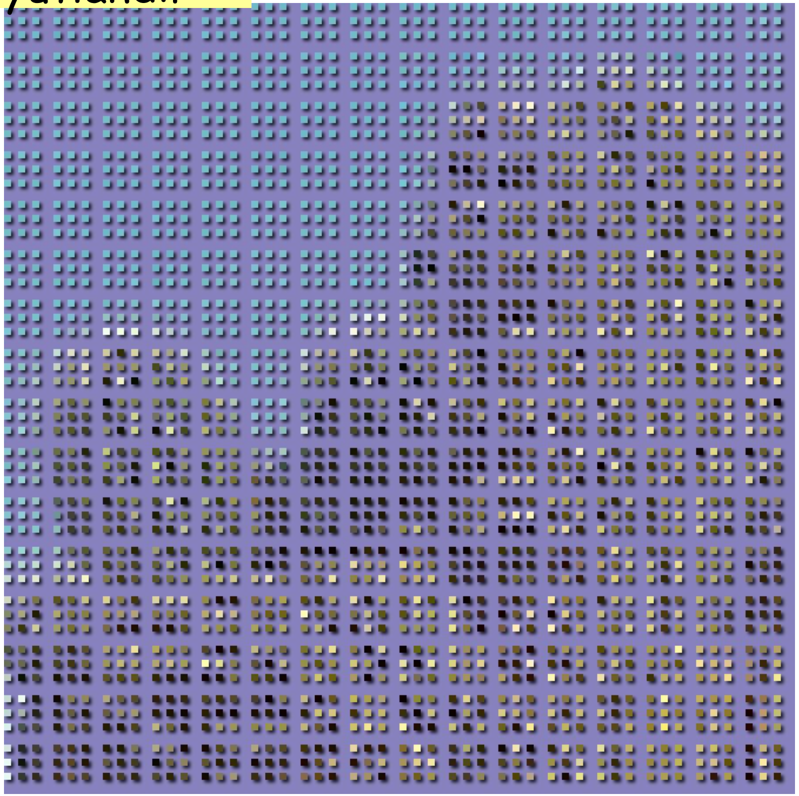
7x7 lik genişlet



En yakın komşu boyutlama

7x7 bloklar
üzerine dağılan
3x3 lük bloklar

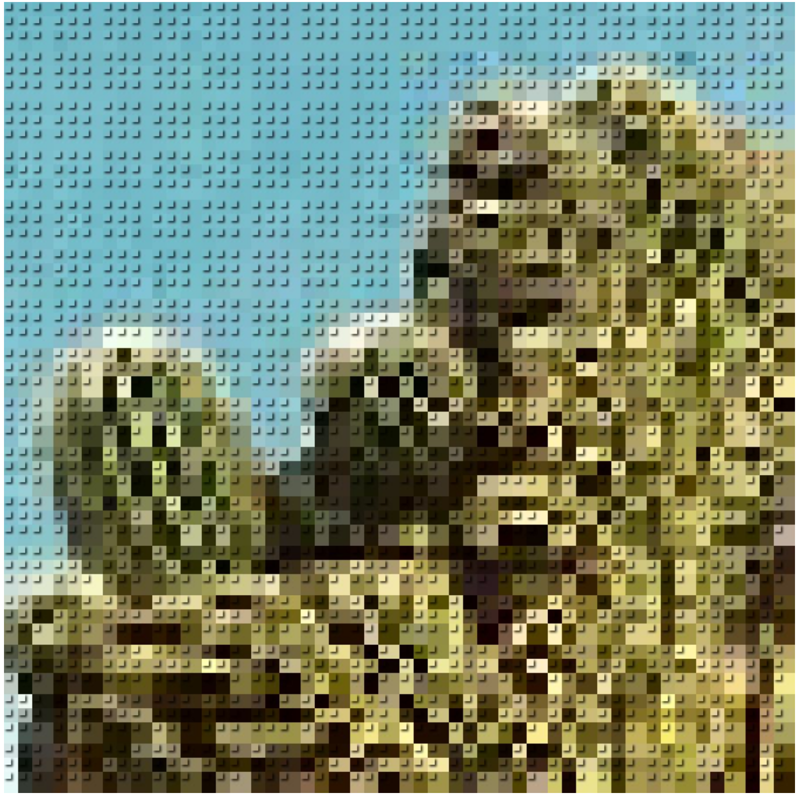
7/3 boyutlandır



7/3 boyutlandır

En yakın komşu boyutlama

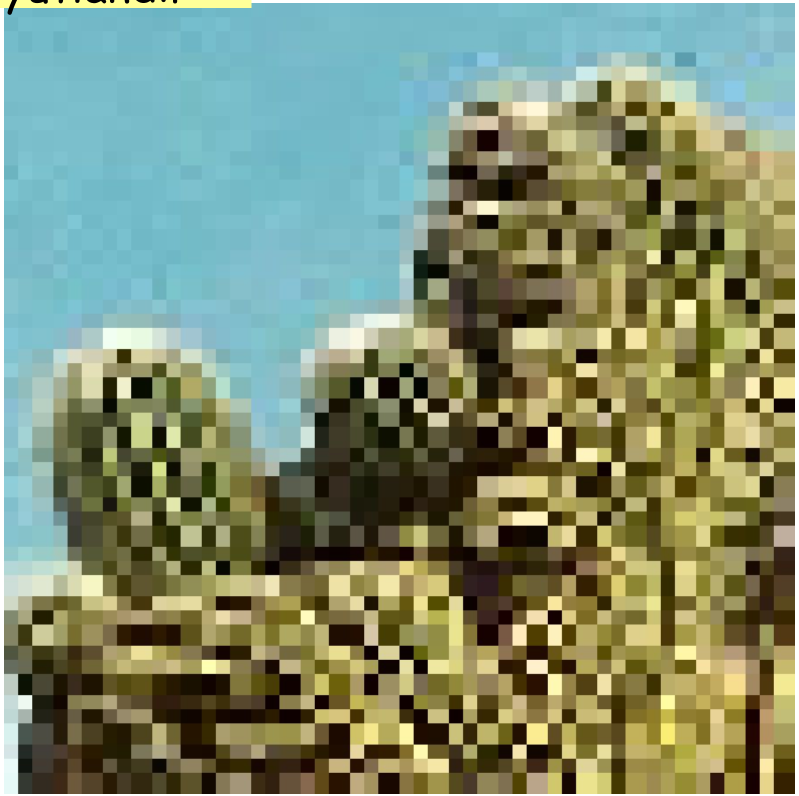
Boş pikseller en yakındaki boş olmayan pikselin rengini alır



En yakın komşu boyutlama

Boş pikseller en yakındaki boş olmayan pikselin rengini alır

7/3 boyutlandır

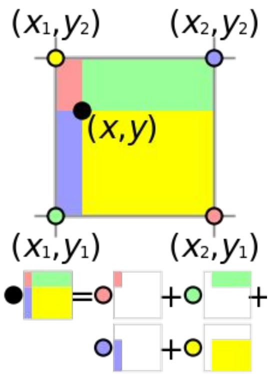


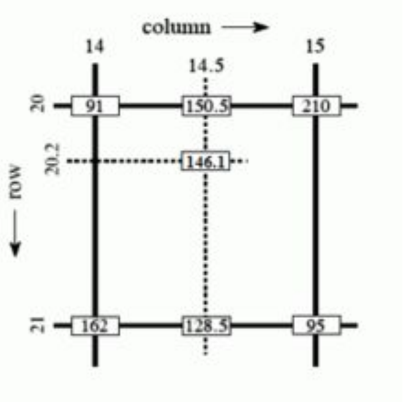
En yakın komşu boyutlama

Orijinal görüntü

7/3 boyutlanmış

Bilinear Interpolation





- Bilinear interpolasyon : $u(x, y) = ax + by + cxy + d$

  $\Rightarrow 4\ bilinmeyen, 4\ denklem$
  - Adım 1) Dört nokta için bu denklem yazılmalı
  - Adım 2) Dört bilinmeyen (a,b,c, d) li dört denklem çözülmeli.
  - Adım 3) Katsayılar (a,b,c,d) hesaplanınca, aynı denklem kullanılarak istenilen x ve y değerlerindeki renk değeri hesaplanabilir.

BiCubic İnterpolasyon örneği

- 16 bilinmeyenli, 16 denklem aşağıdaki gibi inşa edilir ve çözülür.
- Farz edelim ki, (0,0), (1,0), (0,1) ve (1,1) noktalarında *f, fx, fy, fxy (yoğunluk, yatay gradient, dikey gradient, çapraz gradient)* değerleri elimizde. Buna göre 16 bilinmeyenli 16 denklem sistemi aşağıdaki gibi üretilir.

1. $f(0,0) = p(0,0) = a_{00}$
2. $f(1,0) = p(1,0) = a_{00} + a_{10} + a_{20} + a_{30}$
3. $f(0,1) = p(0,1) = a_{00} + a_{01} + a_{02} + a_{03}$
4. $f(1,1) = p(1,1) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij}$

Likewise, eight equations for the derivatives in the x-direction and the y-direction

1. $f_x(0,0) = p_x(0,0) = a_{10}$
2. $f_x(1,0) = p_x(1,0) = a_{10} + 2a_{20} + 3a_{30}$
3. $f_x(0,1) = p_x(0,1) = a_{10} + a_{11} + a_{12} + a_{13}$
4. $f_x(1,1) = p_x(1,1) = \sum_{i=1}^{3} \sum_{j=0}^{3} a_{ij} i$
5. $f_y(0,0) = p_y(0,0) = a_{01}$
6. $f_y(1,0) = p_y(1,0) = a_{01} + a_{11} + a_{21} + a_{31}$
7. $f_y(0,1) = p_y(0,1) = a_{01} + 2a_{02} + 3a_{03}$
8. $f_y(1,1) = p_y(1,1) = \sum_{i=0}^{3} \sum_{j=1}^{3} a_{ij} j$

And four equations for the cross derivative xy.

1. $f_{xy}(0,0) = p_{xy}(0,0) = a_{11}$
2. $f_{xy}(1,0) = p_{xy}(1,0) = a_{11} + 2a_{21} + 3a_{31}$
3. $f_{xy}(0,1) = p_{xy}(0,1) = a_{11} + 2a_{12} + 3a_{13}$
4. $f_{xy}(1,1) = p_{xy}(1,1) = \sum_{i=1}^{3} \sum_{j=1}^{3} a_{ij} i j$

$$p(x,y) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij} x^i y^j .$$

Karşılaştırma