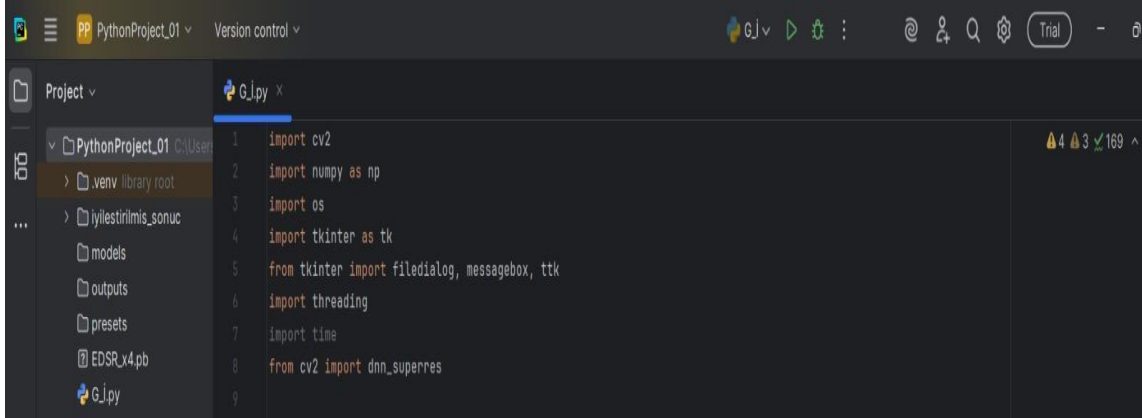


14. Kullanım Kılavuzu

14.1. Kullanılan Kütüphaneler



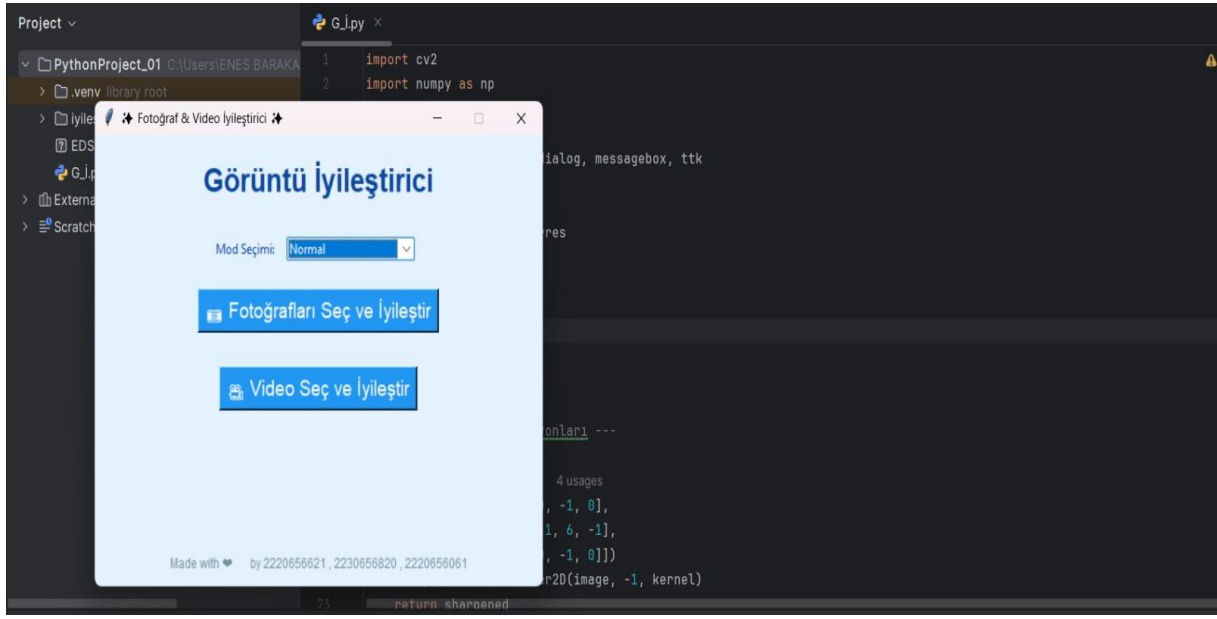
Projemizde kullandığımız kütüphaneler bu şekildedir.

14.2. Proje Kod Yapısı

```
17 # --- İyileştirme Fonksiyonları ---
18
19 def sharpen_image(image): 2 usages
20     """Daha yumuşak ve doğal görünen keskinleştirme efekti"""
21     kernel = np.array([[ -1, -1, -1],
22                        [-1, 9.5, -1],
23                        [-1, -1, -1]])
24     sharpened = cv2.filter2D(image, -1, kernel)
25     return cv2.addWeighted(image, alpha: 0.5, sharpened, beta: 0.5, gamma: 0)
26
27
28 def denoise_image(image): 2 usages
29     """Gelişmiş gürültü azaltma algoritması"""
30     return cv2.fastNlMeansDenoisingColored(image, dst: None, h=8, hColor=8,
31                                           templateWindowSize=7, searchWindowSize=21)
32
33
34 def enhance_contrast(image): 5 usages
35     """Akıllı kontrast artırma"""
36     lab = cv2.cvtColor(image, cv2.COLOR_BGR2LAB)
37     l, a, b = cv2.split(lab)
38     clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8, 8))
39     cl = clahe.apply(l)
40     merged = cv2.merge((cl, a, b))
41     enhanced = cv2.cvtColor(merged, cv2.COLOR_LAB2BGR)
42     return enhanced
43
44
45 def adjust_saturation(image, saturation_factor=1.4): 5 usages
46     """Kontrollü doygunluk artırma"""
47     hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
```

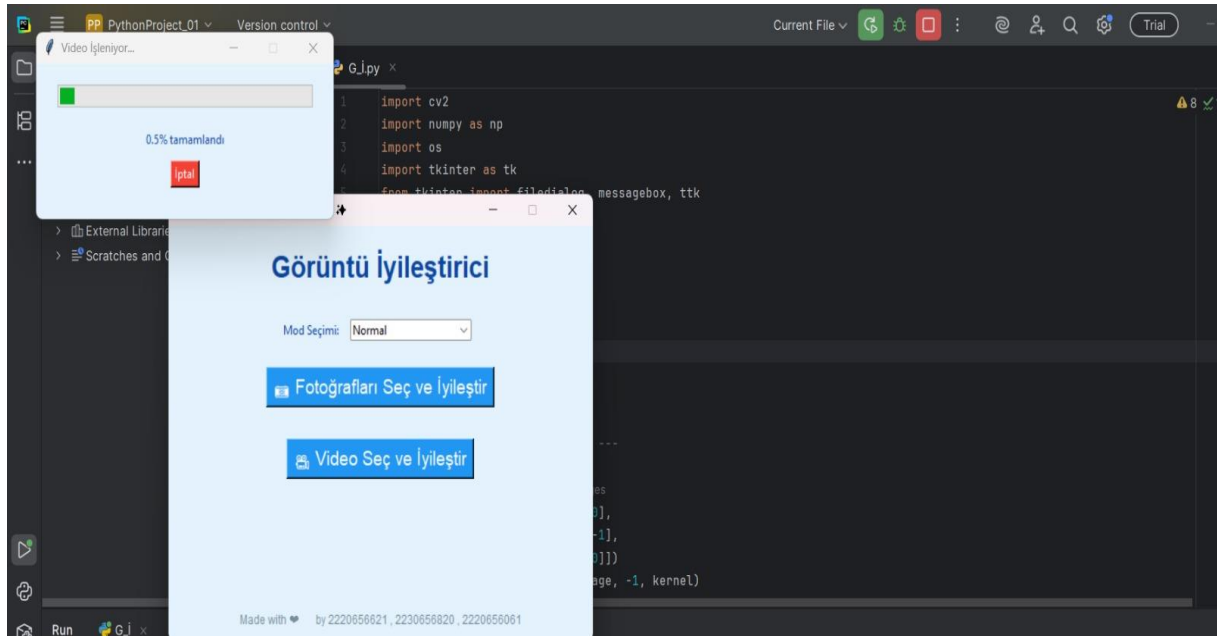
Projemizde kullandığımız iyileştirme fonksiyonları yani fotoğraf ve video iyileştirme kodlarımız bu şekildedir.

14.3. Uygulama Arayüzü



Uygulama arayüzümüz de öncelikle mod seçimi yapılır. Ardından fotoğraf ya da video yüklemesinden hangisi yapılacaksa seçilir, yüklenir ve iyileştirilir.

14.4. Uygulama Arayüzü



Burada ise örnek bir video iyileştirilirken işlenme durumu gösterilmiştir.

14.5. Örnek Fotoğraf İyileştirme



Örnek olarak yapılan fotoğraf iyileştirme.

14.6. Örnek Fotoğraf İyileştirme



Burada ise parlaklık ve düzeltme konusunda bir fotoğraf iyileştirme örneği görülmektedir.