

# MODUL 7

PENGOLAHAN CITRA DIGITAL  
THRESHOLDING CITRA BINER

D3 TEKNIK INFORMATIKA  
JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA  
POLITEKNIK NEGERI BANDUNG



WILDAN SETYA NUGRAHA 032 | PENGOLAHAN CITRA  
DIGITAL | FEBRUARI, 28 2023

---

```
import numpy as np
import pandas as pd
import cv2 as cv
from google.colab.patches import cv2_imshow # for image display
from skimage import io
from PIL import Image
import matplotlib.pyplot as plt
from scipy.ndimage.filters import gaussian_filter
```

```
<ipython-input-31-75c8583e8d36>:8: DeprecationWarning: Please use `gaussian_filter` from the `scipy.ndimage` namespace, the `scipy.ndimage.filters` namespace is depre
from scipy.ndimage.filters import gaussian_filter
```

```
url= 'https://i.pinimg.com/564x/67/dd/6b/67dd6bf95347bd8a5082b2a361b9d554.jpg'
image_0 = io.imread(url)
image_1 = cv.cvtColor(image_0, cv.COLOR_BGR2GRAY)
image_2 = cv.cvtColor(image_0, cv.COLOR_BGR2RGB)
cv2_imshow(image_2)
```



```
def thresholding2(image):
    img = cv.cvtColor(image, cv.COLOR_BGR2GRAY)
    blur = cv.GaussianBlur(img, (5,5), 0)
    ret2, thresh = cv.threshold(blur, 0, 255, cv.THRESH_BINARY_INV + cv.THRESH_OTSU)
    ret, th1 = cv.threshold(img, 127, 255, cv.THRESH_BINARY)
    th2 = cv.adaptiveThreshold(img, 255, cv.ADAPTIVE_THRESH_MEAN_C, cv.THRESH_BINARY, 11, 2)
    th3 = cv.adaptiveThreshold(img, 255, cv.ADAPTIVE_THRESH_GAUSSIAN_C, cv.THRESH_BINARY, 11, 2)
    titles = ['Original Image', 'Global Thresholding (v = 127)', 'Adaptive Mean Thresholding', 'Adaptive Gaussian Thresholding', 'Otsu Thresholding']
    images = [img, th1, th2, th3, thresh]
    fig = plt.figure()
    fig.set_size_inches(10, 10)
    for i in range(5):
        plt.subplot(5, 1, i+1), plt.imshow(images[i], 'gray')
        plt.title(titles[i])
        plt.xticks([], plt.yticks([]))

    plt.show()
```

```
thresholding2(image_0)
```



1. Buat sendiri masing masing fungsi image processing thresholding tersebut (dilarang menggunakan fungsi dari lib populer skimage,opencv,PIL,dll)

Gunakan numpy, matplotlib jika digunakan hanya untuk plot bukan processing

```
def globalThresholding(image, threshold):
    # image grayscale yang dibaca diubah ke dalam array
    image_grayscale = np.array(image)

    # Membuat array kosong yang sama ukurannya dengan array dari image_grayscale
    # hal ini bertujuan untuk nantinya diisi dengan nilai putih dan hitam
    image_binary = np.zeros_like(image_grayscale)

    # Mengganti setiap nilai index dengan logika jika indek di dalam array image_grayscale
    # lebih besar dari threshold atau ambang batas maka nilainya 255
    # tapi jika kurang dari threshold maka nilainya tetap 0
    image_binary[image_grayscale > threshold] = 255

    # Mengembalikan nilai array dari image_binary
    return image_binary
```

```
convertImageToBinary = globalThresholding(image_1, 127)
```

```
Image.fromarray(convertImageToBinary).show()
```

```
plt.imshow(image_1)
```

```
plt.imshow(image_2)
```

```
plt.imshow(image_3)
```

```
plt.imshow(image_4)
```

```
plt.imshow(image_5)
```

```
plt.imshow(image_6)
```

```
plt.imshow(image_7)
```

```
plt.imshow(image_8)
```

```
plt.imshow(image_9)
```

```
plt.imshow(image_10)
```

```
plt.imshow(image_11)
```

```
plt.imshow(image_12)
```

```
plt.imshow(image_13)
```

```
plt.imshow(image_14)
```

```
plt.imshow(image_15)
```

```
plt.imshow(image_16)
```

```
plt.imshow(image_17)
```

```
plt.imshow(image_18)
```

```
plt.imshow(image_19)
```

```
plt.imshow(image_20)
```

```
plt.imshow(image_21)
```

```
plt.imshow(image_22)
```

```
plt.imshow(image_23)
```

```
plt.imshow(image_24)
```

```
plt.imshow(image_25)
```

```
plt.imshow(image_26)
```

```
plt.imshow(image_27)
```

```
plt.imshow(image_28)
```

```
plt.imshow(image_29)
```

```
plt.imshow(image_30)
```

```
plt.imshow(image_31)
```

```
plt.imshow(image_32)
```

```
plt.imshow(image_33)
```

```
plt.imshow(image_34)
```

```
plt.imshow(image_35)
```

```
plt.imshow(image_36)
```

```
plt.imshow(image_37)
```

```
plt.imshow(image_38)
```

```
plt.imshow(image_39)
```

```
plt.imshow(image_40)
```

```
plt.imshow(image_41)
```

```
plt.imshow(image_42)
```

```
plt.imshow(image_43)
```

```
plt.imshow(image_44)
```

```
plt.imshow(image_45)
```

```
plt.imshow(image_46)
```

```
plt.imshow(image_47)
```

```
plt.imshow(image_48)
```

```
plt.imshow(image_49)
```

```
plt.imshow(image_50)
```

```
plt.imshow(image_51)
```

```
plt.imshow(image_52)
```

```
plt.imshow(image_53)
```

```
plt.imshow(image_54)
```

```
plt.imshow(image_55)
```

```
plt.imshow(image_56)
```

```
plt.imshow(image_57)
```

```
plt.imshow(image_58)
```

```
plt.imshow(image_59)
```

```
plt.imshow(image_60)
```

```
plt.imshow(image_61)
```

```
plt.imshow(image_62)
```

```
plt.imshow(image_63)
```

```
plt.imshow(image_64)
```

```
plt.imshow(image_65)
```

```
plt.imshow(image_66)
```

```
plt.imshow(image_67)
```

```
plt.imshow(image_68)
```

```
plt.imshow(image_69)
```

```
plt.imshow(image_70)
```

```
plt.imshow(image_71)
```

```
plt.imshow(image_72)
```

```
plt.imshow(image_73)
```

```
plt.imshow(image_74)
```

```
plt.imshow(image_75)
```

```
plt.imshow(image_76)
```

```
plt.imshow(image_77)
```

```
plt.imshow(image_78)
```

```
plt.imshow(image_79)
```

```
plt.imshow(image_80)
```

```
plt.imshow(image_81)
```

```
plt.imshow(image_82)
```

```
plt.imshow(image_83)
```

```
plt.imshow(image_84)
```

```
plt.imshow(image_85)
```

```
plt.imshow(image_86)
```

```
plt.imshow(image_87)
```

```
plt.imshow(image_88)
```

```
plt.imshow(image_89)
```

```
plt.imshow(image_90)
```

```
plt.imshow(image_91)
```

```
plt.imshow(image_92)
```

```
plt.imshow(image_93)
```

```
plt.imshow(image_94)
```

```
plt.imshow(image_95)
```

```
plt.imshow(image_96)
```

```
plt.imshow(image_97)
```

```
plt.imshow(image_98)
```

```
plt.imshow(image_99)
```

```
plt.imshow(image_100)
```

```
plt.imshow(image_101)
```

```
plt.imshow(image_102)
```

```
plt.imshow(image_103)
```

```
plt.imshow(image_104)
```

```
plt.imshow(image_105)
```

```
plt.imshow(image_106)
```

```
plt.imshow(image_107)
```

```
plt.imshow(image_108)
```

```
plt.imshow(image_109)
```

```
plt.imshow(image_110)
```

```
plt.imshow(image_111)
```

```
plt.imshow(image_112)
```

```
plt.imshow(image_113)
```

```
plt.imshow(image_114)
```

```
plt.imshow(image_115)
```

```
plt.imshow(image_116)
```

```
plt.imshow(image_117)
```

```
plt.imshow(image_118)
```

```
plt.imshow(image_119)
```

```
plt.imshow(image_120)
```

```
plt.imshow(image_121)
```

```
plt.imshow(image_122)
```

```
plt.imshow(image_123)
```

```
plt.imshow(image_124)
```

```
plt.imshow(image_125)
```

```
plt.imshow(image_126)
```

```
plt.imshow(image_127)
```

```
plt.imshow(image_128)
```

```
plt.imshow(image_129)
```

```
plt.imshow(image_130)
```

```
plt.imshow(image_131)
```

```
plt.imshow(image_132)
```

```
plt.imshow(image_133)
```

```
plt.imshow(image_134)
```

```
plt.imshow(image_135)
```

```
plt.imshow(image_136)
```

```
plt.imshow(image_137)
```

```
plt.imshow(image_138)
```

```
plt.imshow(image_139)
```

```
plt.imshow(image_140)
```

```
plt.imshow(image_141)
```

```
plt.imshow(image_142)
```

```
plt.imshow(image_143)
```

```
plt.imshow(image_144)
```

```
plt.imshow(image_145)
```

```
plt.imshow(image_146)
```

```
plt.imshow(image_147)
```

```
plt.imshow(image_148)
```

```
plt.imshow(image_149)
```

```
plt.imshow(image_150)
```

```
plt.imshow(image_151)
```

```
plt.imshow(image_152)
```

```
plt.imshow(image_153)
```

```
plt.imshow(image_154)
```

```
plt.imshow(image_155)
```

```
plt.imshow(image_156)
```

```
plt.imshow(image_157)
```

```
plt.imshow(image_158)
```

```
plt.imshow(image_159)
```

```
plt.imshow(image_160)
```

```
plt.imshow(image_161)
```

```
plt.imshow(image_162)
```

```
plt.imshow(image_163)
```

```
plt.imshow(image_164)
```

```
plt.imshow(image_165)
```

```
plt.imshow(image_166)
```

```
plt.imshow(image_167)
```

```
plt.imshow(image_168)
```

```
plt.imshow(image_169)
```

```
plt.imshow(image_170)
```

```
plt.imshow(image_171)
```

```
plt.imshow(image_172)
```

```
plt.imshow(image_173)
```

```
plt.imshow(image_174)
```

```
plt.imshow(image_175)
```

```
plt.imshow(image_176)
```

```
plt.imshow(image_177)
```

```
plt.imshow(image_178)
```

```
plt.imshow(image_179)
```

```
plt.imshow(image_180)
```

```
plt.imshow(image_181)
```

```
plt.imshow(image_182)
```

```
plt.imshow(image_183)
```

```
plt.imshow(image_184)
```

```
plt.imshow(image_185)
```

```
plt.imshow(image_186)
```

```
plt.imshow(image_187)
```

```
plt.imshow(image_188)
```

```
plt.imshow(image_189)
```

```
plt.imshow(image_190)
```

```
plt.imshow(image_191)
```

```
plt.imshow(image_192)
```

```
plt.imshow(image_193)
```

```
plt.imshow(image_194)
```

```
plt.imshow(image_195)
```

```
plt.imshow(image_196)
```

```
plt.imshow(image_197)
```

```
plt.imshow(image_198)
```

```
plt.imshow(image_199)
```

```
plt.imshow(image_200)
```

```
plt.imshow(image_201)
```

```
plt.imshow(image_202)
```

```
plt.imshow(image_203)
```

```
plt.imshow(image_204)
```

```
plt.imshow(image_205)
```

```
plt.imshow(image_206)
```

```
plt.imshow(image_207)
```

```
plt.imshow(image_208)
```

```
plt.imshow(image_209)
```

```
plt.imshow(image_210)
```

```
plt.imshow(image_211)
```

```
plt.imshow(image_212)
```

```
plt.imshow(image_213)
```

```
plt.imshow(image_214)
```

```
plt.imshow(image_215)
```

```
plt.imshow(image_216)
```

```
plt.imshow(image_217)
```

```
plt.imshow(image_218)
```

```
plt.imshow(image_219)
```

```
plt.imshow(image_220)
```

```
plt.imshow(image_221)
```

```
plt.imshow(image_222)
```

```
plt.imshow(image_223)
```

```
plt.imshow(image_224)
```

```
plt.imshow(image_225)
```

```
plt.imshow(image_226)
```

```
plt.imshow(image_227)
```

```
plt.imshow(image_228)
```

```
plt.imshow(image_229)
```

```
plt.imshow(image_230)
```

```
plt.imshow(image_231)
```

```
plt.imshow(image_232)
```

```
plt.imshow(image_233)
```

```
plt.imshow(image_234)
```

```
plt.imshow(image_235)
```

```
plt.imshow(image_236)
```

```
plt.imshow(image_237)
```

```
plt.imshow(image_238)
```

```
plt.imshow(image_239)
```

```
plt.imshow(image_240)
```

```
plt.imshow(image_241)
```

```
plt.imshow(image_242)
```

```
plt.imshow(image_243)
```

```
plt.imshow(image_244)
```

```
plt.imshow(image_245)
```

```
plt.imshow(image_246)
```

```
plt.imshow(image_247)
```

```
plt.imshow(image_248)
```

```
plt.imshow(image_249)
```

```
plt.imshow(image_250)
```

```
plt.imshow(image_251)
```

```
plt.imshow(image_252)
```

```
plt.imshow(image_253)
```

```
plt.imshow(image_254)
```

```
plt.imshow(image_255)
```

```
plt.imshow(image_256)
```

```
plt.imshow(image_257)
```

```
plt.imshow(image_258)
```

```
plt.imshow(image_259)
```

```
plt.imshow(image_260)
```

```
plt.imshow(image_261)
```

```
plt.imshow(image_262)
```

```
plt.imshow(image_263)
```

```
plt.imshow(image_264)
```

```
plt.imshow(image_265)
```

```
plt.imshow(image_266)
```

```
plt.imshow(image_267)
```

```
plt.imshow(image_268)
```

```
plt.imshow(image_269)
```

```
plt.imshow(image_270)
```

```
plt.imshow(image_271)
```

```
plt.imshow(image_272)
```

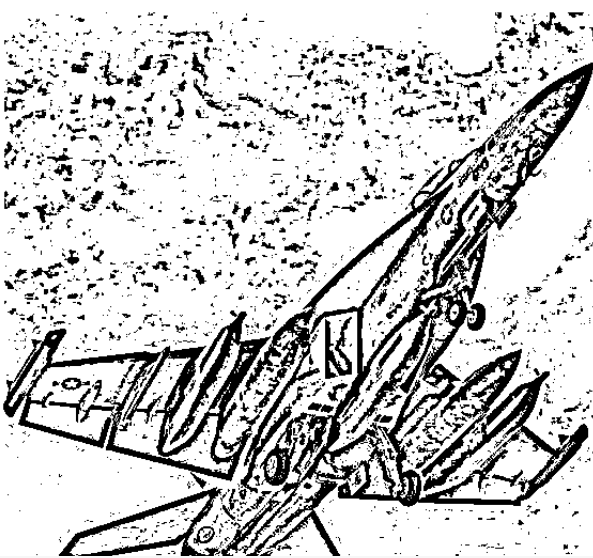
```
plt.imshow(image_273)
```

```
plt.imshow(image_274)
```

```
plt.imshow(image_275)
```

```
plt.imshow(image_276)
```

```
plt.imshow(image_277)
```



```
def adaptive_threshold_gaussian(img, block_size, c):
    # Memasukkan nilai height dan width dalam sebuah citra yang diinputkan
    h, w = img.shape

    # Membuat sebuah array kosong dimana array ini akan memudahkan dalam mengganti
    # nilai menjadi hitam atau putih. Ukuran array ini akan sama dengan ukuran
    # citra yang diinputkan
    output = np.zeros_like(img)

    img_padded = np.pad(img, block_size // 2, mode='constant', constant_values=0)

    for y in range(h):
        for x in range(w):
            block = img_padded[y:y+block_size, x:x+block_size]

            # Bedanya pada adaptive gaussian threshold dengan adaptive
            # mean threshold, di adaptive gaussian threshold itu di block-block
            # yang udah ditentukan akan digunakan filter gaussian terlebih dahulu
            block_smoothed = gaussian_filter(block, sigma=block_size / 6)

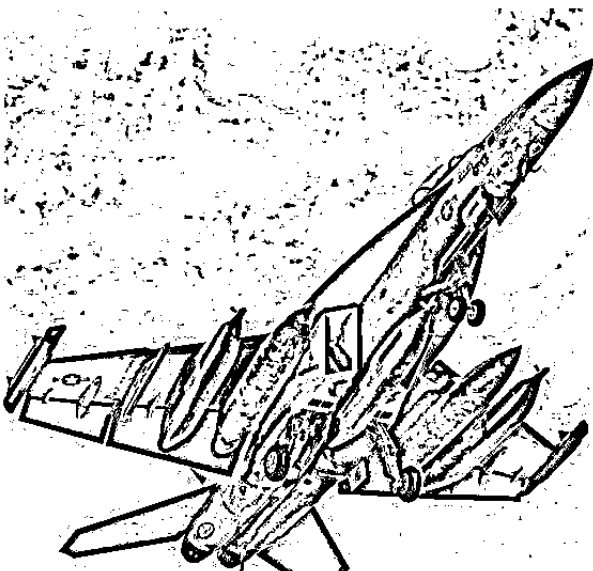
            # Menghitung nilai rata-rata dan dikurang dengan konstanta untuk
            # menghasilkan nilai threshold pada suatu block tertentu
            threshold = np.mean(block_smoothed) - c

            # Untuk mengganti nilai setiap indeksnya maka diperlukan logika
            # dimana logikan tersebut akan mengecek setiap nilai piksel di dalam
            # block tertentu apakah nilainya melebihi threshold atau tidak
            # jika melebihi maka diganti menjadi nilai 255 (Putih), jikalau
            # tidak maka diganti 0 (Hitam)
            if img[y, x] >= threshold:
                output[y, x] = 255
            else:
                output[y, x] = 0

    return output
```

```
imgGaussianThreshold = adaptive_threshold_gaussian(image_1, 11, 2)
```

```
Image.fromarray(imgGaussianThreshold).show()
```



```
def otsu_threshold(img_gray):
    # Hitung histogram citra grayscale
    hist, bins = np.histogram(img_gray, bins=256, range=(0, 255))

    # Normalisasi histogram
    hist_norm = hist.astype('float') / np.sum(hist)

    # Hitung probabilitas kelas
    class_prob = np.cumsum(hist_norm)
```

```

# Hitung rata-rata kelas
class_mean = np.cumsum(hist_norm * np.arange(0, 256))

# Hitung rata-rata global
global_mean = np.sum(hist_norm * np.arange(0, 256))

# Hitung variansi dalam kelas
class_var = np.zeros_like(hist_norm)
for i in range(1, 256):
    w0 = class_prob[i]
    w1 = 1 - w0
    u0 = class_mean[i] / w0
    u1 = (global_mean - class_mean[i]) / w1
    class_var[i] = w0 * w1 * (u0 - u1)**2

# Cari nilai threshold yang meminimalkan variansi dalam kelas
threshold = np.argmax(class_var)

# Konversi citra grayscale menjadi citra biner menggunakan threshold hasil Otsu
img_binary = np.zeros_like(img_gray)
img_binary[img_gray >= threshold] = 255

return img_binary

```

```
imgOtsuThreshold = otsu_threshold(image_1)
```

```
Image.fromarray(imgOtsuThreshold).show()
```





# Lesson Learn PCD Pertemuan 7

No  
Date

Nama : Wildan setya nugraha

NIM : 211511032

What went well ?

Dari pertemuan kali ini saya belajar mengenai threshold citra atau dengan kata lain mengubah citra menjadi citra biner

citra biner itu nilai pikselnya hanya 0 dan 1

untuk mengubah ke bentuk citra biner ini ada beberapa cara

1. Global Thresholding

2. Adaptive Mean Thresholding

3. Adaptive Gaussian Thresholding

4. Otsu thresholding

Untuk memudahkan melakukan perubahan ke citra biner, alangkah baiknya mengubah citra berwarna

ke citra grayscale hanya memiliki 1 kanal atau 8 bit

Threshold ini merupakan nilai ambang, yang dimana sebagai penentu nilai piksel itu bernilai 0 atau 255.

1. Global Threshold

ini merupakan cara yg paling mudah karena kita tinggal tentukan nilai thresholdnya lalu menggunakan np.array () untuk mengubah citra menjadi ke dalam array. Setelah itu setiap piksel akan di cek nilainya

Jika nilai piksel > threshold, maka ubah menjadi 255, selain itu ubah nilainya jadi 0



## 2. Adaptive Threshold mean

ini caranya pada array citra kita bagi ke dalam block-block. ukuran block bisa kita tentukan sendiri.

Dari setiap block yang sudah ditentukan, kita tinggal hitung mean dari setiap block menggunakan np.mean.

hasil mean ini akan menjadi threshold. Namun harus dikurangi konstanta terlebih dahulu.

Setelah itu caranya sama seperti global threshold

## 3. Adaptive Gaussian Threshold.

ini caranya sama saja. Namun yg membedakan adalah setiap block itu di filter gaussian terlebih dahulu, lalu dihitung meannya. dan jangan lupa untuk mengurangnya dgn konstanta. Setelah itu caranya sama dgn global threshold.

## 4. Otsu threshold

ini cara untuk mengetahui thresholdnya adalah dengan membuat histogramnya terlebih dahulu setelah itu, menghitung class variansinya

dgn cara  $V_b = W_1 W_2 (\mu_1 - \mu_2)^2$

dimana :  $W_1$  = Probabilitas piksel yg termasuk kelas background

$W_2$  = Probabilitas piksel yg termasuk kelas foreground

$\mu_1$  = rata-rata kelas background

$\mu_2$  = rata-rata kelas foreground.



- What didn't go well?

kesulitan mengilustrasikan pada cara mencari threshold menggunakan otsu.

- What might have been better handled if done differently?

mengilustrasikan prosesnya terlebih dahulu.

boleh mencobanya dan ukuran array tidak terlalu besar

- What recommendations would you give to others who might be involved in future projects of a similar type?

1. Memahami fungsi-fungsi selain dari yang diberikan Open cv
2. Mengubah citra menjadi citra grayscale dulu
3. ilustasikan proses
4. jika ingin cepat mengilustrasikannya boleh meminta bantuan chatGpt.