Python code using OpenCV to remove all colors apart from black and replace them with white:

Python

import cv2

import numpy as np

# Load the image

img = cv2.imread('input\_image.jpg')

# Convert the image to grayscale

gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

# Threshold the grayscale image to get only black pixels

\_, thresh = cv2.threshold(gray, 50, 255, cv2.THRESH\_BINARY\_INV)

# Create a 3-channel image with white pixels where threshold is 255

white\_img = cv2.merge([255-thresh, 255-thresh, 255-thresh])

# Output the resulting image

cv2.imwrite('output\_image.jpg', white\_img)

# Display the output image

cv2.imshow('Output Image', white\_img)

cv2.waitKey(0)

cv2.destroyAllWindows()

This code works by:

1. Loading the input image
2. Converting the image to grayscale
3. Thresholding the grayscale image to get a binary image with black pixels (value 0) and non-black pixels (value 255)
4. Creating a 3-channel image with white pixels (value 255) where the threshold is 255 (non-black pixels)
5. Saving and displaying the output image

Note:

* You can adjust the threshold value (50) to include different shades of black.
* This code assumes that "black" means pixel values close to (0, 0, 0). If your image has dark gray or dark brown pixels that you want to consider "black", adjust the threshold value accordingly.

Replace 'input\_image.jpg' with the path to your input image.