Now, our model is ready to be tested with real scenarios. Below is the sample code for using OpenCV to access our webcam and predicting whether each frame contains fire or not. If a frame contains fire in it, we want to change the color of that frame to B&W.

import cv2  
import numpy as np  
from PIL import Image  
import tensorflow as tf  
from keras.preprocessing import image#Load the saved model  
model = tf.keras.models.load\_model('InceptionV3.h5')  
video = cv2.VideoCapture(0)while True:  
 \_, frame = video.read()#Convert the captured frame into RGB  
 im = Image.fromarray(frame, 'RGB')#Resizing into 224x224 because we trained the model with this image size.  
 im = im.resize((224,224))  
 img\_array = image.img\_to\_array(im)  
 img\_array = np.expand\_dims(img\_array, axis=0) / 255  
 probabilities = model.predict(img\_array)[0]  
 #Calling the predict method on model to predict 'fire' on the image  
 prediction = np.argmax(probabilities)  
 #if prediction is 0, which means there is fire in the frame.  
 if prediction == 0:  
 frame = cv2.cvtColor(frame, cv2.COLOR\_RGB2GRAY)  
 print(probabilities[prediction])cv2.imshow("Capturing", frame)  
 key=cv2.waitKey(1)  
 if key == ord('q'):  
 break  
video.release()  
cv2.destroyAllWindows()

