

- (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3, No. 7, 2012, pp. 40-43.
2. Духнич Е.И., Демищенко Д.А. Модификация шифра Фейстеля с инволютивной кватернионной матрицей // Вестник ГМУ им. адмирала Ф.Ф. Ушакова. – 2020. – №3 (32). – С.92-96.
  3. Mariusz Dzwonkowski, Michal Papaj, and Roman Rykaczewski. A New Quaternion-Based Encryption Method for DICOM Images. IEEE Transactionsonimageprocessing, vol. 24, no. 11, November 2015, pp. 4614-4622.
  4. Ключарев П.Г. О статическом тестировании блочных шифров. // Математика и математическое моделирование – 2018 – №5 – С. 35-36.
  5. <https://webhome.phy.duke.edu/~rgb/General/dieharder.php>
  6. <https://www.cryptool.org/en/>
- References**
1. Sastry V. U. K, K. Anup Kumar, A Modified Feistel Cipher Involving ModularArithmetic Addition and Modular Arithmetic Inverse of a Key Matrix. (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3, No. 7, 2012, pp. 40-43.
  2. E.I. Duhnich, D.A. Demishchenko. Modifikaciya shifra Fejstelya s involyutivnojkvaternionnoj matricей. Vestnik GMU im. Admirala F.F. Ushakova.- 2020. -№3 (32).-S.92-96.
  3. Mariusz Dzwonkowski, Michal Papaj, and Roman Rykaczewski. A New Quaternion-Based Encryption Method for DICOM Images. IEEE Transactionsonimageprocessing, vol. 24, no. 11, November 2015, pp. 4614-4622.
  4. Klyucharyov P.G. O sticheskom testirovanii blochnyh shifrov. Matematika i matematicheskoe modelirovanie- 2018 №5. С. 35-36.
  5. <https://webhome.phy.duke.edu/~rgb/General/dieharder.php>
  6. <https://www.cryptool.org/en/>

УДК 004.032.2

DOI: 10.34046/aumsuomt100/25

## СРАВНЕНИЕ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ОБНАРУЖЕНИЯ МЕДИЦИНСКОЙ МАСКИ НА ФОТО И ВИДЕО

*Г. Е. Панамарев, доктор технических наук, профессор  
И.В. Родыгина, кандидат технических наук, доцент  
А.А. Шевченко, магистрант*

Во время пандемии COVID-19 во всём мире приняты рекомендации к ношению медицинских масок и ограничен допуск в места массового скопления людей тех, кто игнорирует это требование. С помощью свёрточных нейронных сетей и глубокого обучения есть возможность анализировать видеопоток, чтобы облегчить работу на пропускных пунктах и отмечать людей без масок. В данной статье рассматриваются три нейронные сети, их обучение и сравнительный анализ. Данный пример показывает, как можно легко подготовить нейронную сеть под определённую задачу.

**Ключевые слова:** нейронная сеть, глубокое обучение, CNN, свёрточная сеть, компьютерное зрение, обучение нейронных сетей, анализ видеопотока, MobileNet V2, Xception, Inception V3

## COMPARISON OF NEURAL NETWORKS FOR DETECTING A MEDICAL MASK ON PHOTOS AND VIDEOS

*G.E. Panamarev, I.V. Rodygina, A.A. Shevchenko*

During the COVID-19 pandemic, recommendations for wearing medical masks have been adopted around the world and admission to crowded places of those who ignore this requirement is limited. With the help of convolutional neural networks and deep learning, it is possible to analyze the video stream to facilitate work at checkpoints and mark people without masks. This article examines three neural networks, their training and comparative analysis. This example shows how you can easily prepare a neural network for a specific task.

**Keywords:** neural network, deep learning, CNN, machine learning, computer vision, MobileNet V2, Xception, Inception V3.

**Введение.** Болезнь, вызванная коронавирусной инфекцией COVID-2019, – это инфекционное заболевание, вызванное новым, ранее неизвестным коронавирусом. У большинства заболевших COVID-19 наблюдаются легкие или умеренные симптомы, выздоровление происходит без специфического лечения [2].

Одной из мер защиты от распространения COVID-19 является ношение медицинской маски, поэтому во всех общественных местах требуют выполнять эту рекомендацию. Для этого на входах в торговые центры, метро, транспорт и т.д. стоят пропускные пункты, где измеряют температуру и проверяют наличие защитных средств. Но,

если учитывать человеческий фактор и халатность, то в общественные места могут проходить люди без защитных средств и тем самым подвергать опасности как себя, так и других людей.

**Постановка задачи.** Разработать систему, которая будет анализировать видеопоток и выдавать предупреждение об отсутствии маски у человека и отмечать его, давая тем самым возможность сделать ему предупреждение.

Для осуществления поставленной задачи и обучения нейронной сети мы будем использовать фреймворк машинного обучения Tensorflow 2 [1].

Задачу можно разбить на несколько этапов:

- Чтение видеопотока и разбиение его на фреймы, преобразование каждого фрейма подпараметры входного слоя нейронной сети
- Обнаружение наличия лиц на входном изображении и получение координат
- Каждое найденное лицо пропустить через заранее обученную нейронную сеть и, в зависимости от предсказания, поставить метку наличия или отсутствия маски
- Отобразить полученный результат

В качестве детектора лиц возьмем уже предобученную модель. А нейронную сеть построим на основе нескольких скомпилированных моделей из фреймворка Tensorflow 2 и произведём сравнительный анализ.

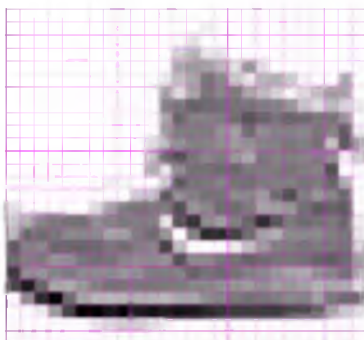
Для основы нейронной сети возьмём следующие модели:

- MobileNetV2
- Xception
- InceptionV3

### Принцип работы свёрточных нейронных сетей при распознавании объектов на изображении

Для того, чтобы можно было распознать наличие какого-либо объекта на изображении, необходимо выделить повторяющиеся свойства. Для этого изображение RGB рассматривается отдельно в каждом спектре (red-green-blue) и пропускается через фильтра, так называемые

#### After ReLU Processing (26x26)



свертки. Принцип работы свёрточных слоев заключается в выделении более значимых признаков на выбранном участке изображения (ядро выборки) путём перемножения каждого значения на весовые коэффициенты, рис. 1 [4].

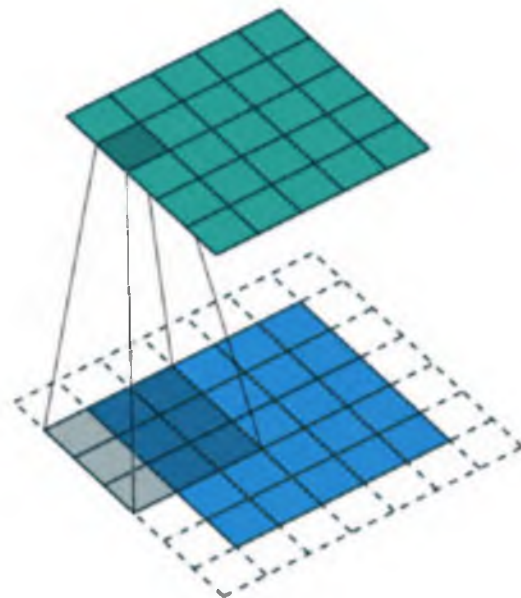


Рисунок 1 – Пример операции свёртки

Ещё одной основной операцией в свёрточных нейронных сетях является операция подвыборки или max-pooling. Принцип работы данного слоя заключается в прохождении исходного изображения «окном» заданного размера, и нахождении более выраженных пикселей, т.е. всё «окно» заменяется одним значением, таким образом получается изображение меньшего размера, но сохранившее все свойства, рис. 2.

Чтобы выделить схожие свойства на изображениях, операции свёртки и подвыборки чередуют, применяя различные входные параметры и функции активации. В конце нейронной сети последний слой трансформируется в одномерный массив и после этого применяется один или несколько полносвязных слоёв для получения массива размерностью с количеством классов, которые нужно предсказать, рис. 3.

#### After Max Pooling (13x13)

Stride = 2 Pixels  
Filter Size = 2x2 Pixels

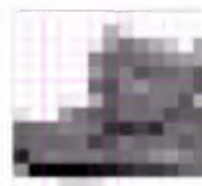


Рисунок 2 – Пример операции подвыборки



Рисунок 3 – Принцип работы полносвязной нейронной сети

**Обучение нейронной сети.** Для обучения нейронной сети загрузим датасет «FaceMaskDetection» с сайта Kaggle[5]. Он содержит около 12 тыс. изображений, разбитых на 2 категории, рис. 4.

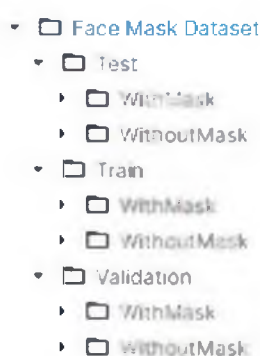


Рисунок 4 – Структура файлов датасета «FaceMaskDetection»

Для того, чтобы можно было получить более точные результаты о качестве обучения сети, нужно разделить данные, на которых будет учиться сеть, от данных, которые будут подаваться для проверки. Другими словами, для проверки нужно дать данные, которые нейронная сеть видит первый раз.

MobileNetV2 – разработана командой Google и позиционирует себя как легковесная нейронная сеть для работы на мобильных устройствах, является усовершенствованной версией MobileNetV1, показала примерно одинаковую точность при большей скорости обработки изображений, рис. 5 [6, 7]. Отличительной особенностью от предыдущей версии является наличие

свёртки 1x1 с линейной функцией активации, которая понижает число каналов. Такой блок свёртки называют bottlenecklayer.

Input	Operator
$224^2 \times 3$	conv2d
$112^2 \times 32$	bottleneck
$112^2 \times 16$	bottleneck
$56^2 \times 24$	bottleneck
$28^2 \times 32$	bottleneck
$14^2 \times 64$	bottleneck
$14^2 \times 96$	bottleneck
$7^2 \times 160$	bottleneck
$7^2 \times 320$	conv2d 1x1
$7^2 \times 1280$	avgpool 7x7
$1 \times 1 \times 1280$	conv2d 1x1

Рисунок 5 – Архитектура MobileNetV2

После обучения нейронной сети MobileNetV2 на выбранном датасете можно увидеть, что точность на тестовых данных составила 99,8%, рис. 6.

После проверки работы сети на нескольких изображениях можно увидеть, что она допустила несколько ошибок. При этом одна ошибка на объекте малого размера, ещё одна на неполном изображении лица и две на лицах, где маски есть, но они спущены, рис. 7.

**Xception.** Архитектура Inception была предложена в 2015 году. Идея заключалась в том, чтобы вместо выбора размер ядра, взять не-



сколько вариантов сразу, использовать их все одновременно и конкатенировать результаты. Однако это существенно увеличивает количество операций, которые необходимо выполнить для вычисления активаций одного слоя, поэтому со-

здатели предлагают такую хитрость: перед каждым свёрточным блоком делать свёртку с размером ядра 1x1, снижая размерность сигнала, подающегося на вход свёрткам с большими размерами ядер [8].

```

320/320 [=====] - 20s 39ms/step - loss: 0.0334 - accuracy: 0.9887 - val_loss: 0.0164 - val_accuracy: 0.9950
Epoch 2/5
320/320 [=====] - 11s 35ms/step - loss: 0.0039 - accuracy: 0.9989 - val_loss: 0.0023 - val_accuracy: 0.9975
Epoch 3/5
320/320 [=====] - 11s 35ms/step - loss: 0.0017 - accuracy: 0.9998 - val_loss: 0.0022 - val_accuracy: 0.9990
Epoch 4/5
320/320 [=====] - 11s 35ms/step - loss: 0.0011 - accuracy: 0.9996 - val_loss: 0.0020 - val_accuracy: 0.9975
Epoch 5/5
320/320 [=====] - 11s 35ms/step - loss: 5.5372e-04 - accuracy: 1.0000 - val_loss: 0.0026 - val_accuracy: 0.9980
[INFO] evaluating network...
63/63 [=====] - 3s 32ms/step - loss: 0.0076 - accuracy: 0.9980
    
```

Рисунок 6—обучение MobileNetV2



Рисунок 7— Проверка работы обученной сети MobileNetV2

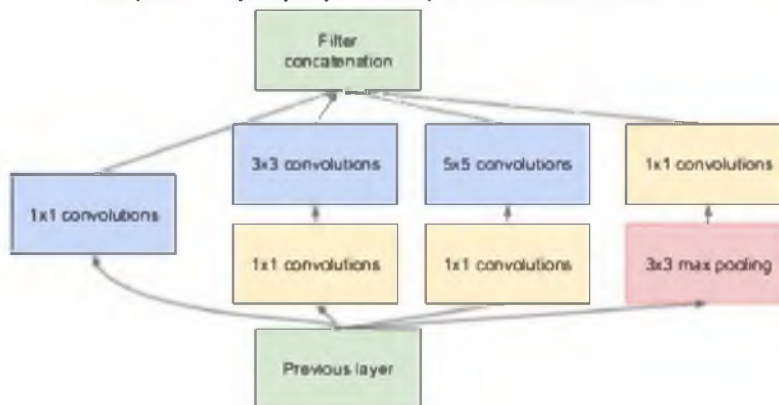


Рисунок 8 –Модуль Inception

В 2016 году был предложен модуль экстремальный Inception, также известный как depthwise separable convolution, который работает следующим образом:

- Pointwise convolution - свертка исходного тензора фильтром 1x1, как это делается в блоке Inception.

• Depthwise spatial convolution - каждый канал сворачивается по отдельности свёрткой 3x3 (при этом размерность не изменится, так как сворачиваются не все каналы вместе, как в обычном свёрточном слое).

После обучения Xception на выбранном датасете можно увидеть, что точность на тестовых данных составила 99,75%, рис. 10.

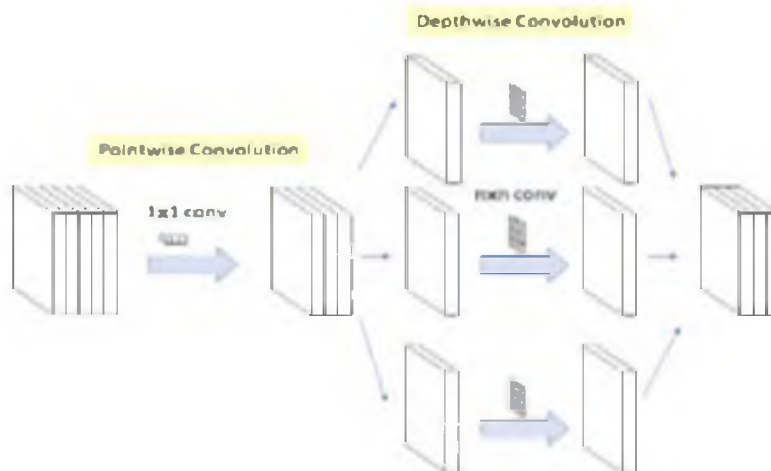


Рисунок 9 – Модуль Xception

```

328/328 [.....] - 28s 66ms/step - loss: 0.0046 - accuracy: 0.9855 - val_loss: 0.0184 - val_accuracy: 0.9965
Epoch 2/5
326/328 [.....] - 28s 62ms/step - loss: 0.0177 - accuracy: 0.9967 - val_loss: 0.0176 - val_accuracy: 0.9968
Epoch 3/5
328/328 [.....] - 28s 62ms/step - loss: 0.0122 - accuracy: 0.9966 - val_loss: 0.0095 - val_accuracy: 0.9988
Epoch 4/5
328/328 [.....] - 28s 62ms/step - loss: 0.0094 - accuracy: 0.9970 - val_loss: 0.0067 - val_accuracy: 0.9988
Epoch 5/5
328/328 [.....] - 28s 61ms/step - loss: 0.0065 - accuracy: 0.9984 - val_loss: 0.0053 - val_accuracy: 0.9975
[INFO] evaluating network ...
63/63 [.....] - 6s 67ms/step - loss: 0.0053 - accuracy: 0.9975
    
```

Рисунок 10 – обучение Xception

После проверки работы сети на нескольких изображениях можно увидеть, что она допустила

несколько ошибок. Сеть распознала объект малого размера, но неполное изображение и спущенные пропустила так же, как и MobileNetV2.



Рисунок 11 – Проверка работы обученной сети Xception



**Inceptionv3.** Из-за того, что расположение искомого объекта на изображении может находиться в разных местах и масштабах, выбор размера ядра становится трудной задачей. Более большое ядро подходит для объектов большего размера, в то время как маленькое ядро обнаружит мелкие предметы. Чем глубже нейронная сеть, тем она более подвержена переобучению, имеет трудности с

распространением градиента и требует больших вычислительных ресурсов для обучения [10].

Команда GoogLeNet предложила архитектуру Inception, где используются фильтры разных размеров на одном уровне, рис. 12. Таким образом сеть стала шире, а не глубже.

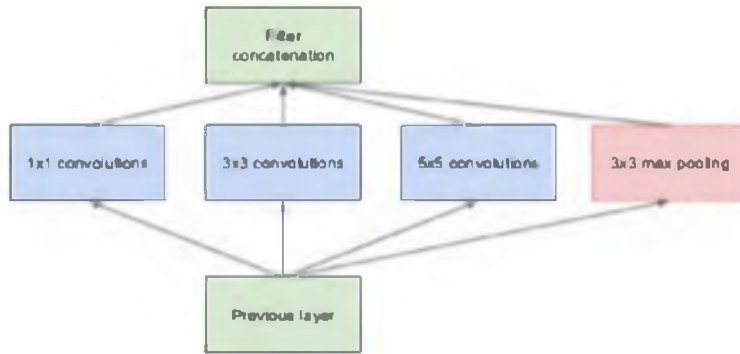


Рисунок 12 – Модуль InceptionV1

Для того, чтобы уменьшить потребление вычислительных ресурсов, была введена дополнительная свертка 1x1 перед свертками 3x3 и 5x5. Таким образом была полезна сеть InceptionV1.

Нейронные сети работают лучше, когда свертки не сильно меняют размеры входных данных. Слишком сильное уменьшение размеров может вызвать потерю информации, известную как «representationalbottleneck».

Используя интеллектуальные методы факторизации, свертки можно сделать более эффективными с точки зрения вычислительной сложности.

В версии InceptionV2 было предложено разложить свертку 5x5 на две операции свертки 3x3, чтобы повысить скорость вычислений, рис. 13. Хотя это может показаться нелогичным, свертка 5x5 в 2.78 раза «дороже» свертки 3x3. Таким образом, наложение двух свертки 3x3 фактически приводит к повышению производительности.

Более того, были разложены свертки с размером фильтра NxN на комбинацию свертки 1xN и Nx1. Например, свертка 3x3 эквивалентна сначала выполнению свертки 1x3, а затем выполнению свертки 3x1 на ее выходе. Было обнаружено, что этот метод на 33% «дешевле», чем одинарная свертка 3x3, рис. 14.

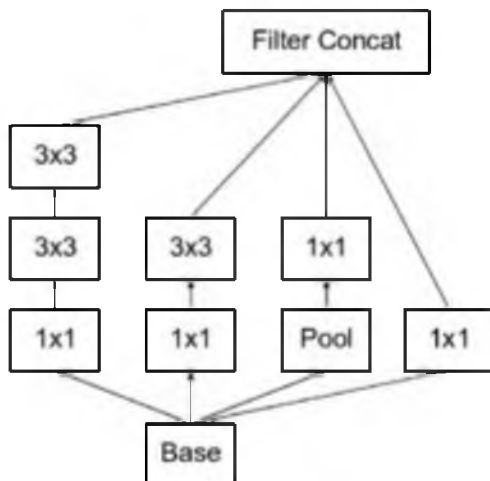


Рисунок 13 – Разложение свертки 5x5 на 3x3 в модуле InceptionV2

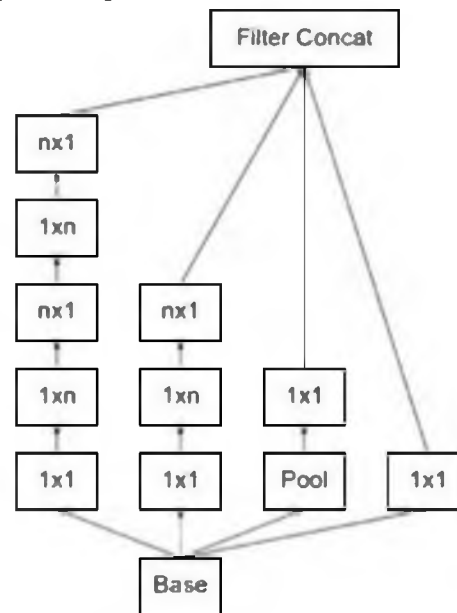


Рисунок 14 – Разложение свертки NxN на комбинацию свертки 1xN и Nx1в модуле InceptionV2

Блоки фильтров в модуле были расширены (сделаны шире, а не глубже), чтобы устранить репрезентативное узкое место. Если вместо этого сделать модуль глубже, произойдет чрезмерное уменьшение размеров и, как следствие, потеря информации, рис. 15.

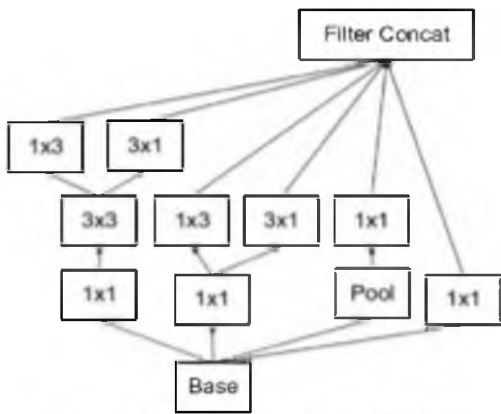


Рисунок 15 – Расширение блоков фильтра

В версии InceptionV3 были внесены следующие обновления:

- Оптимизатор RMSProp



Рисунок 16 – Архитектура Inception V3

```

1/0/320 [-----] - 38s 67ms/step - loss: 0.0282 - accuracy: 0.9898 - val_loss: 0.0051 - val_accuracy: 0.9995
Epoch 2/5
320/320 [-----] - 28s 61ms/step - loss: 0.0056 - accuracy: 0.9998 - val_loss: 0.0043 - val_accuracy: 0.9995
Epoch 3/5
320/320 [-----] - 28s 63ms/step - loss: 0.0015 - accuracy: 0.9996 - val_loss: 0.0039 - val_accuracy: 0.9995
Epoch 4/5
320/320 [-----] - 28s 62ms/step - loss: 5.3271e-04 - accuracy: 1.0000 - val_loss: 0.0024 - val_accuracy: 0.9995
[INFO] evaluating network...
63/63 [-----] - 6s 50ms/step - loss: 0.0024 - accuracy: 0.9995
    
```

Рисунок 17 – обучение InceptionV3



Рисунок 18 – Проверка работы обученной сети Inception V3

- Факторизованные свертки 7x7
- BatchNorm во вспомогательных классификаторах
- Сглаживание меток (тип регуляризирующего компонента, добавляемого в формулу потерь, который не позволяет сети стать слишком уверенной в отношении класса. Предотвращает чрезмерную подгонку), рис. 16.

После обучения InceptionV3 на выбранном датасете можно увидеть, что точность на тестовых данных составила 99,95%. см. рис. 17.

После проверки работы сети на нескольких изображениях можно увидеть, что сеть справилась лучше, чем предыдущие, но сделала 2 ошибки –на размытом изображении и при спущенной маске, см. рис. 18.

**Распознавание объектов в видеопотоке.**

После того, как обучили нейронную сеть распознавать наличие или отсутствие маски на изображении, нужно сделать возможным распознавание из видеопотока. Для этого воспользуемся библиотекой `imutils` и разобьем видеопоток на фреймы, т.е. изображения каждого кадра. Требуется обра-

ботать каждый кадр из видеопотока: найти позиции лиц на кадре и подать каждое из них обученной нейронной сети, алгоритм обработки представлен на рисунке 19. После этого нарисовать рамки найденных лиц, поставить метку с наличием или отсутствием маски и отобразить полученный результат.

Таблица 1 – Сравнение результатов работы нейронные сетей

Нейронная сеть	Скорость распознавания, RTX 3070/GTX 1650 ti, мс/кадр	Точность, %	Кол-во ложных срабатываний	Размер модели, Мб
MobileNet V2	32/65	99.8	4/31	10,8
Xception	67/201	99.75	3/31	85
InceptionV3	58/131	99.95	2/31	87,2

```
# loop over the frames from the video stream
while True:
    # grab the frame from the threaded video stream and resize it
    # to have a maximum width of 400 pixels
    frame = vs.read()
    frame = imutils.resize(frame, width=400)

    # detect faces in the frame and determine if they are wearing a
    # face mask or not
    (locs, preds) = detect_and_predict_mask(frame, faceNet, maskNet)

    # loop over the detected face locations and their corresponding
    # locations
    for (box, pred) in zip(locs, preds):
        # unpack the bounding box and predictions
        (startX, startY, endX, endY) = box
        (mask, withoutMask) = pred

        # determine the class label and color we'll use to draw
        # the bounding box and text
        label = "Mask" if mask > withoutMask else "No Mask"
        color = (0, 255, 0) if label == "Mask" else (0, 0, 255)

        # include the probability in the label
        label = "{}: {:.2f}%".format(label, max(mask, withoutMask) * 100)

        # display the label and bounding box rectangle on the output
        # frame
        cv2.putText(frame, label, (startX, startY - 10),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.45, color, 2)
        cv2.rectangle(frame, (startX, startY), (endX, endY), color, 2)

    # show the output frame
    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF

    # if the `q` key was pressed, break from the loop
    if key == ord("q"):
        break
```

Рисунок 19 – Код для обработки видеопотока



**Заключение.** Тестирование проводилось на видеокарте RTX 3070, и GTX 1650 ti, которые имеют 5888 и 1024 шейдерных блоков соответственно, см. таблицу 1. Из полученных результатов можно сделать вывод, что самой точной и с меньшим количеством ошибок оказалась нейронная сеть InceptionV3, но она более требовательна к ресурсам, т.к. занимает 87.2 Мб и обрабатывает 1 кадр за 58 мс. Если нам важна скорость для работы на более слабых, или мобильных устройствах, то выбор однозначно падает на MobileNetV2.

#### Литература:

1. Фреймворк Tensorflow 2. URL: <https://www.tensorflow.org/> (дата обращения 28.07.2021).
2. Основные сведения о COVID-19 URL: <https://www.who.int/ru/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronaviruse-disease-covid-19> (дата обращения 28.07.2021).
3. Глубокое обучение: разбираемся со свертками. URL: <https://medium.com/@balovbohdan/глубокое-обучение-разбираемся-со-свертками-6e47bfc27792> (дата обращения 29.07.2021).
4. Введение в сверточные нейронные сети (Convolutional Neural Networks). URL: <https://habr.com/ru/post/454986/> (дата обращения 29.07.2021).
5. Face Mask Detection ~12K Images Dataset. URL: <https://www.kaggle.com/ashishjangra27/face-mask-12k-images-dataset> (дата обращения 29.07.2021).
6. MobileNet: меньше, быстрее, точнее. URL: <https://habr.com/ru/post/352804/> (дата обращения 29.07.2021).
7. MobileNetV2: инвертированные остатки и линейные узкие места. URL: <https://www.machinelearningmastery.ru/mobilenetv2-inverted-residuals-and-linear-bottlenecks-8a4362f4ffd5/> (дата обращения 30.07.2021).
8. Xception: компактная глубокая нейронная сеть. URL: <https://habr.com/ru/post/347564/> (дата обращения 30.07.2021).
9. Обзор Xception. URL: <https://www.machinelearningmastery.ru/review-xception-with-depthwise-separable-convolution-better-than-inception-v3-image-dc967dd42568/> (дата обращения 30.07.2021).
10. A Simple Guide to the Versions of the Inception Network. URL: <https://towardsdatascience.com/a-simple-guide-to-the-versions-of-the-inception-network-7fc52b863202> (дата обращения 31.07.2021).

#### References

1. FrejmworkTensorflow 2. URL: <https://www.tensorflow.org/> (data obrashcheniya 28.07.2021).
2. Osnovnyesvedeniya o COVID-19 URL: <https://www.who.int/ru/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronaviruse-disease-covid-19> (data obrashcheniya 28.07.2021).
3. Glubokoeobuchenie: razbiraemysya so svertkami. URL: <https://medium.com/@balovbohdan/glubokoe-obuchenie-razbiraemysya-so-svertkami-6e47bfc27792> (data obrashcheniya 29.07.2021).
4. Vvedenie v svyortochnyenejromysetsi (Convolutional Neural Networks). URL: <https://habr.com/ru/post/454986/> (data obrashcheniya 29.07.2021).
5. Face Mask Detection ~12K Images Dataset. URL: <https://www.kaggle.com/ashishjangra27/face-mask-12k-images-dataset> (data obrashcheniya 29.07.2021).
6. MobileNet: men'she, bystree, tochnee. URL: <https://habr.com/ru/post/352804/> (data obrashcheniya 29.07.2021).
7. MobileNetV2: invertirovannyeostatki-kiilinejnyeuuzkiemesta. URL: <https://www.machinelearningmastery.ru/mobilenetv2-inverted-residuals-and-linear-bottlenecks-8a4362f4ffd5/> (data obrashcheniya 30.07.2021).
8. Xception: kompaktnayaglubokayanejronnaya set'. URL <https://habr.com/ru/post/347564/> (data obrashcheniya 30.07.2021).
9. ObzorXception. URL: <https://www.machinelearningmastery.ru/review-xception-with-depthwise-separable-convolution-better-than-inception-v3-image-dc967dd42568/> (data obrashcheniya 30.07.2021).
10. A Simple Guide to the Versions of the Inception Network. URL: <https://towardsdatascience.com/a-simple-guide-to-the-versions-of-the-inception-network-7fc52b863202> (data obrashcheniya 31.07.2021).