

# Дигитално процесирање на слика

Тема:

Детекција на ирис, визуелни карактеристики за ирис и споредба

## Вовед

Детекцијата и споредбата на ирисот претставуваат напреден биометриски метод за идентификација кој ги искористува уникатните карактеристики на ирисот на човечкото око за процесот на автентикација. Со примена на модели за препознавање на објекти во слики, прво се детектира ирисот на сликата од окото, а потоа се користат различни техники за генерирање на дескриптори на ирисот, кои се применуваат за споредба на две слики од ириси и проценка на нивната сличност.

Во овој проект ќе искористиме различни модели и дескриптори и ќе направиме споредба помеѓу нивната ефикасност и перформанси во детектирање и споредба на две слики од ирис.

## Модел

За детекција на ирис ќе користиме два модели, MobileNetV2 и VGG16. За работа со моделите ја користиме бесплатната библиотека Keras којашто е API на високо ниво за платформата Tensorflow и ни нуди полесен начин за работа со вештачка интелигенција.

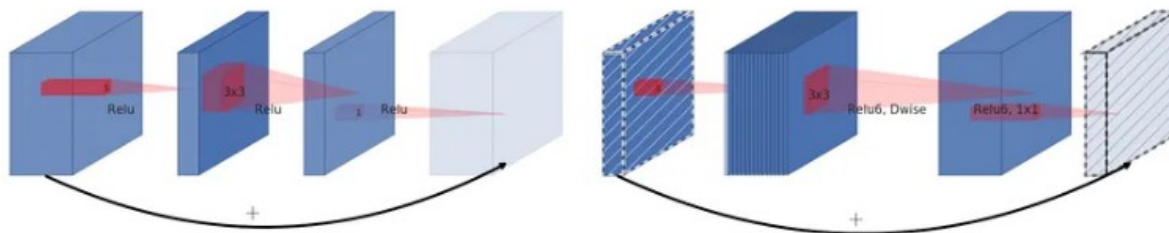
Го користиме податочното множество OpenEDS (Open Eye Dataset) достапен на овој [линк](#) со лабелите достапни [овде](#). Бидејќи ова податочно множество во оригинална форма е составено од повеќе од 300.000 слики коишто би одземале долго време за обработување со достапните спецификации, користиме случајно избрани 9301 слика од кои 90% за тренирање и 10% за тестирање на моделот.

## MobileNetV2 модел

MobileNetV2 е тип на невронска мрежа развиена од Google која вметнува нови иновативни идеи во развојот на мрежата за разлика од нејзиниот предок MobileNetV1 кои и овозможуваат да има подобри перформанси и ефикасност. Дизајнирана е да ја намали комплексноста на математичките пресметки, што ја прави идеална за мобилни уреди или системи со ограничени компјутерски ресурси. Таа се користи главно за задачи како класификација на слики и детекција на објекти.

Позади оваа мрежа има 3 главни идеи:

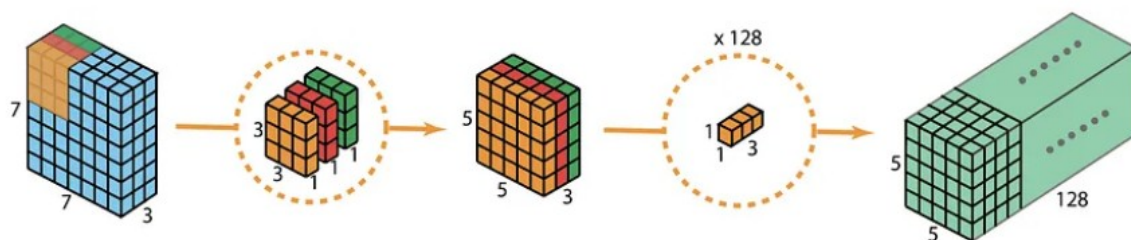
- Инвертирани резидуални блокови – на влез прима блок со ниска димензија кој по примена на  $1 \times 1$  конволуциски слој со Relu6 активациона функција добива поголема димензија. Потоа применува  $3 \times 3$  длабочински базиран конволуциски слој со Relu6 активација кој извлекува важни информации за сликата и на крај користејќи уште еден  $1 \times 1$  конволуциски слој со линеарна активација блокот се враќа во почетната димензија. За да се овозможи побрз проток при backpropagation се додава врска помеѓу првиот и последниот слој



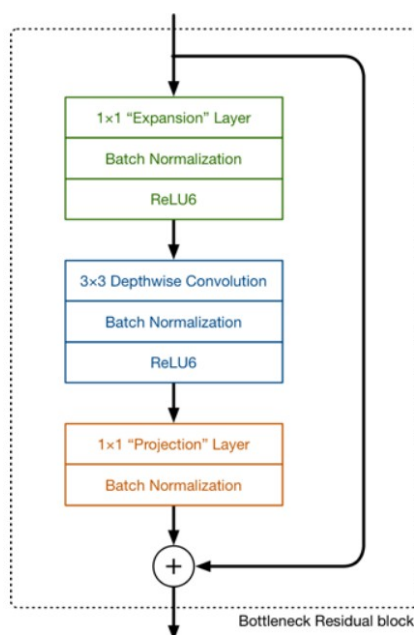
Сл. 1 Споредба помеѓу традиционалниот резидуален слој (лево) и инвертиран резидуален слој (десно)

- Линеарни тесни точки - специјални слоеви со мал број на канали помеѓу два поголеми слоја кои користат лесни линеарни пресметки за да се намали комплексноста, но ги задржуваат важните информации

- Разделив конволуциски слој - го дели традиционалниот конволуциски слој на два дела, прво секој филтер се применува на секој канал од сликата, а потоа се применува  $1 \times 1$  конволуција на сите канали заедно. На овој начин се намалува бројот на пресметките и се овозможува побрза обработка на податоците



Сл. 2 Пример за разделив конволуциски слој со влез  $7 \times 7 \times 3$  и излез  $5 \times 5 \times 128$



Сл. 3 Подетален приказ на разделив конволуциски слој

MobileNetV2 невронската мрежа на влез прима серија од слики со димензија  $224 \times 224$  во 3 канали. Нејзината архитектура е главно составена од инвертираните резидуални блокови. Составена е од следните слоеви:

- Влезен Conv2D слој со големина на влез  $224 \times 224 \times 3$
- Инвертиран резидуален слој со големина на влез  $112 \times 112 \times 32$
- Инвертиран резидуален слој со големина на влез  $112 \times 112 \times 16$
- Инвертиран резидуален слој со големина на влез  $56 \times 56 \times 24$
- Инвертиран резидуален слој со големина на влез  $28 \times 28 \times 32$
- Инвертиран резидуален слој со големина на влез  $14 \times 14 \times 64$
- Инвертиран резидуален слој со големина на влез  $14 \times 14 \times 96$
- Инвертиран резидуален слој со големина на влез  $7 \times 7 \times 160$
- Conv2D слој со големина на влез  $7 \times 7 \times 320$
- average pooling  $7 \times 7$
- Conv2D слој со големина на влез  $1 \times 1 \times 1280$

average pooling  $7 \times 7$  е слој којшто ја зема средната вредност од вредностите во матрица со димензии  $7 \times 7$ .

Нашиот модел е изграден од следните слоеви:

- MobileNetV2 кој има влез со димензии  $224 \times 224 \times 3$  и го оневозможуваме неговото тренирање бидејќи е претходно трениран на ImageNet
- GlobalAveragePooling2D
- Dense слој со 128 неврони и активациска функција ReLU
- Dense слој со 64 неврони и активациска функција ReLU
- Dense слој со 16 неврони и активациска функција ReLU
- Dense слој со 3 неврони за излезите (x,y координати на центарот и радиусот на ирисот)

За подготовка на податочното множество за тренирање на моделот сликите ги претвораме во димензии од  $224 \times 224$  и потоа ги нормализираме во вредности помеѓу 0 и 1. Потоа сликите и лабелите ги претвораме во numpy низи и ги предаваме на моделот.

За тренирање на моделот користиме оптимизатор adam со предодредената стапка на учење од 0.001. За функција на загуба го користиме средноквадратното растојание помеѓу вистинските и предвидените вредности. За да се намали веројатноста од overfitting користиме регуларизација од тип на early stopping, којашто ја следи валидациската загуба и кога во три епохи по ред валидациската загуба станува полоша од претходната, моделот се враќа на епохата пред да почне да се влошува валидациската загуба и тука завршува тренирањето. Исто така при тренирањето користиме големина на batch од 32, а за валидација ги користиме последните 10% од податоците од тренирачкото множество.

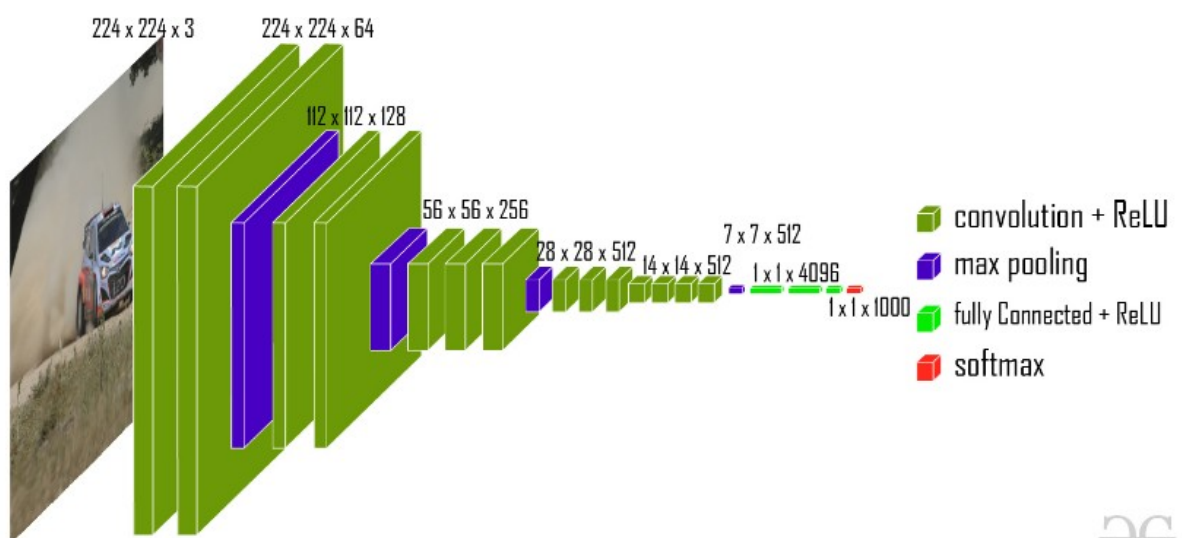
Моделот заврши со тренирање после 36 епохи со точност од 98.60% на тест множеството. Има 2.777.931 вкупен број на параметри од кои 173.315 тренирачки, 2.257.984 нетренирачки и 346.632 параметри на оптимизаторот.

## VGG16 модел

VGG16 е длабока конволуциска мрежа која е карактеристична по нејзината длабочина. Составена е од 16 слоеви, од кои 13 конволуциски слоеви и 3 целосно поврзани слоеви. Моделот е познат по неговите добри перформанси во класифицирање на слики, со точност од 92.7% на податочното множество на ImageNet што го вметнува во листата на најдобрите 5 модели тренирани на тоа множество. Но, за разлика од MobileNetV2, овој модел не е изграден со комплексноста на ум, поради што му треба повеќе време за тренирање на истите податоци и со истите додадени слоеви во споредба со MobileNetV2.

Овој модел е составен од следните слоеви:

- Влезен слој со димензии на влез  $224 \times 224 \times 3$
- Два конволуциски слоеви со  $64 \ 3 \times 3$  филтри
- Max Pooling слој со блок  $2 \times 2$
- Два конволуциски слоеви со  $128 \ 3 \times 3$  филтри
- Max Pooling слој со блок  $2 \times 2$
- Два конволуциски слоеви со  $256 \ 3 \times 3$  филтри
- Два сета од три конволуциски слоеви со  $512 \ 3 \times 3$  филтри
- Max Pooling слој со блок  $2 \times 2$
- Два дополнителни конволуциски слоеви со  $3 \times 3$  филтри
- Flatten слој
- Три целосно поврзани слоја од кои првите два со ReLU активација и последниот излезен слој со Softmax активација



Сл. 4 Архитектурата на VGG16

Нашиот модел е изграден од следните слоеви:

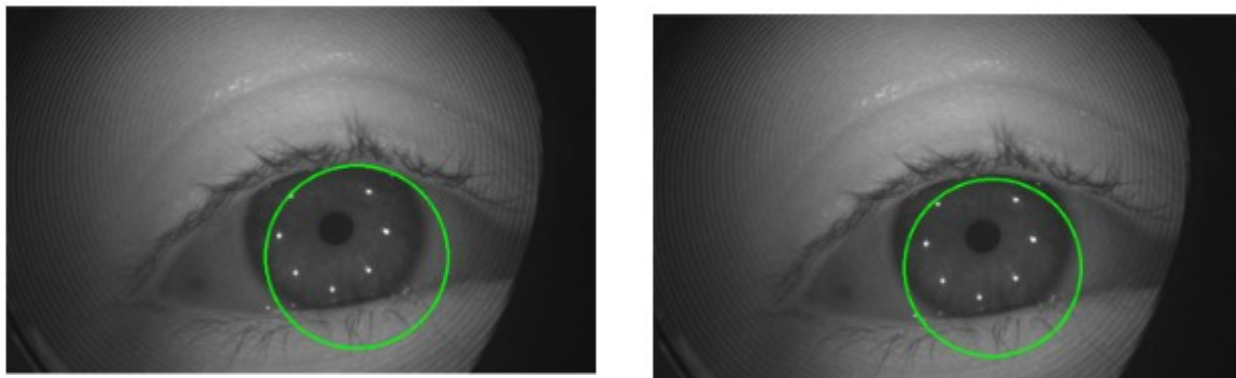
- VGG16 кој има димензии на влез  $224 \times 224 \times 3$ , и го оневозможуваме неговото тренирање бидејќи претходно е трениран на ImageNet
- Flatten слој којшто ја намалува димензионалноста
- Dense слој со 128 неврони и активациска функција ReLU
- Dense слој со 64 неврони и активациска функција ReLU
- Dense слој со 16 неврони и активациска функција ReLU
- Dense слој со 3 неврони за излезите (x,y координати на центарот и радиусот на ирисот)

Подготовката на податочното множество за тренирање и тренирањето на моделот во следниот чекор е извршено со истите параметри како и тренирањето на моделот MobileNetV2 опишано претходно.

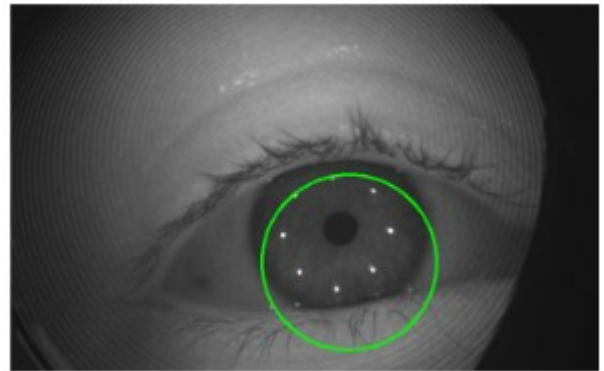
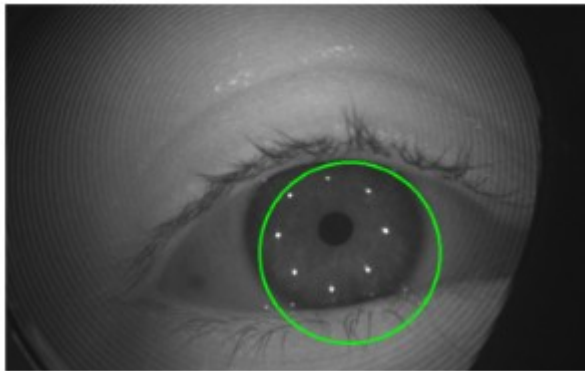
Овој модел заврши со тренирање после 31 епохи со точност од 99.18% на тест множеството. Има вкупно 17.935.429 број на параметри од кои 3.220.739 тренирачки, 14.714.688 нетренирачки и 2 параметри на оптимизаторот.

## Користење на моделот за детекција на ирисот

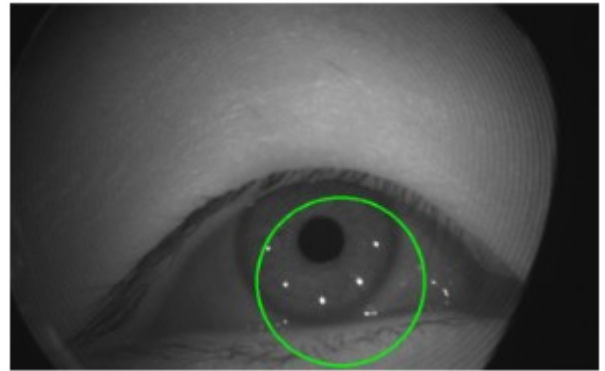
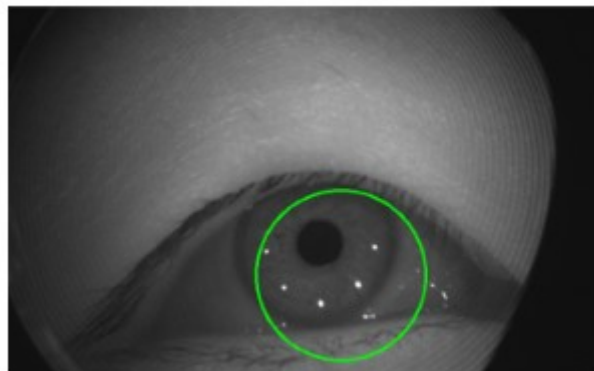
Пред да го искористиме тренираниот модел, прво сликата ја трансформираме во потребниот облик. Ја претвараеме со димензии  $224 \times 224$  и ја нормализираме, а потоа и додаваме димензија за симулирање на испраќање во batch за да може да ја обработи моделот којшто на влез прима само серии од податоци. Како резултат од моделот добиваме предвидени x и y координати на центарот на ирисот и неговиот радиус. Ја претвораме сликата во grayscale бидејќи како таква ќе ја користиме во два дескриптори и потоа креираме маска со димензии на сликата. Користејќи го резултатот од моделот, цртаме круг на маската којшто го претставува регионот на ирисот кој потоа сакаме да го задржиме. Користејќи го bitwise\_and методот од OpenCV ги задржуваме само деловите од сликата коишто со моделот се предвидени дека се совпаѓаат со ирисот, а останатиот дел од сликата ќе остане црн. Добиената слика ја претвораме со димензии  $224 \times 224$ .



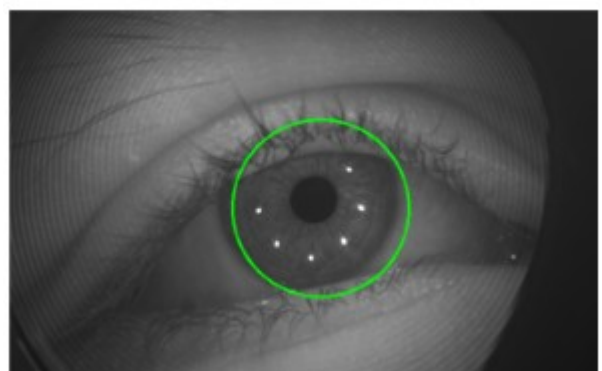
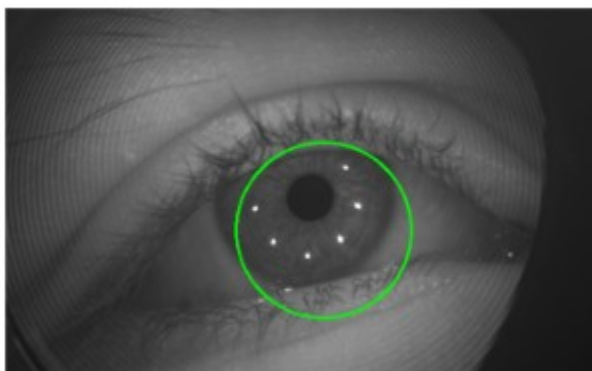
Сл. 5 Споредба на детекција на ирис со MobileNetV2 (лево) и VGG16 (десно)



Сл. 6 Споредба на детекција на ирис со MobileNetV2 (лево) и VGG16 (десно)

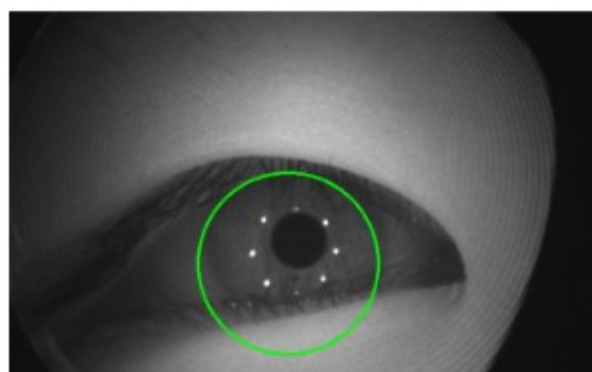
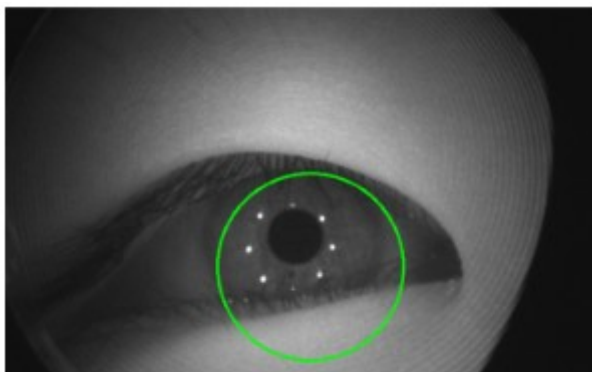


Сл. 7 Споредба на детекција на ирис со MobileNetV2 (лево) и VGG16 (десно)

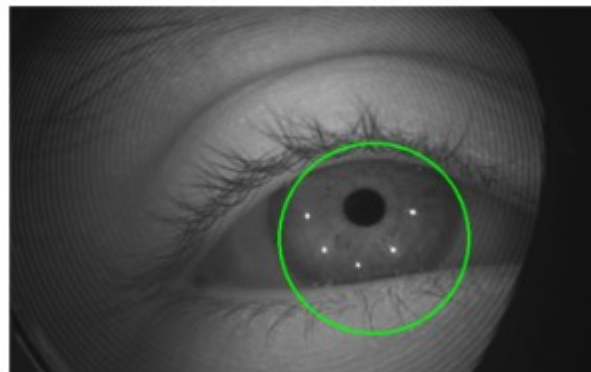
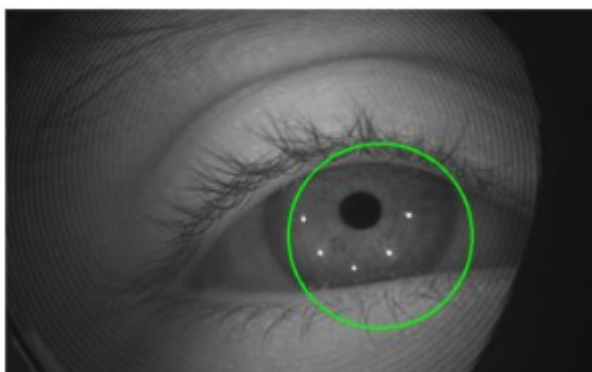


Сл. 8 Споредба на детекција на ирис со MobileNetV2 (лево) и VGG16 (десно)

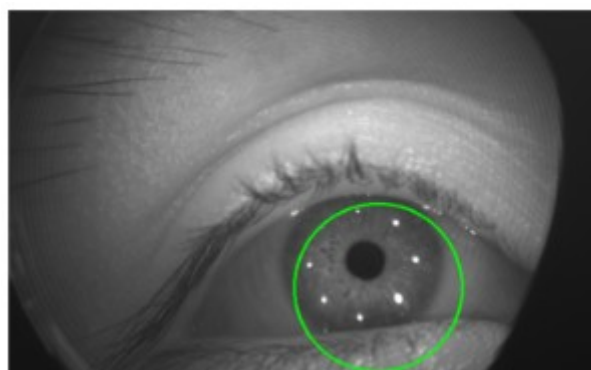
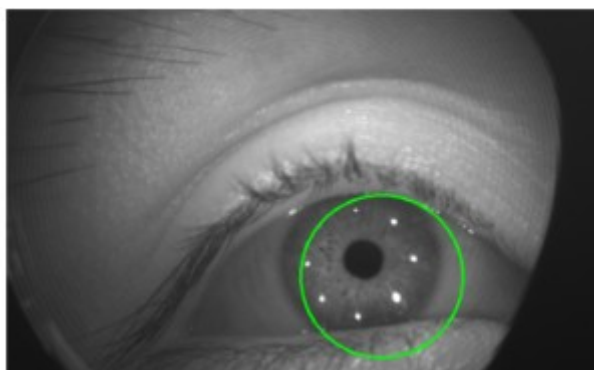




Сл. 9 Споредба на детекција на ирис со MobileNetV2 (лево) и VGG16 (десно)

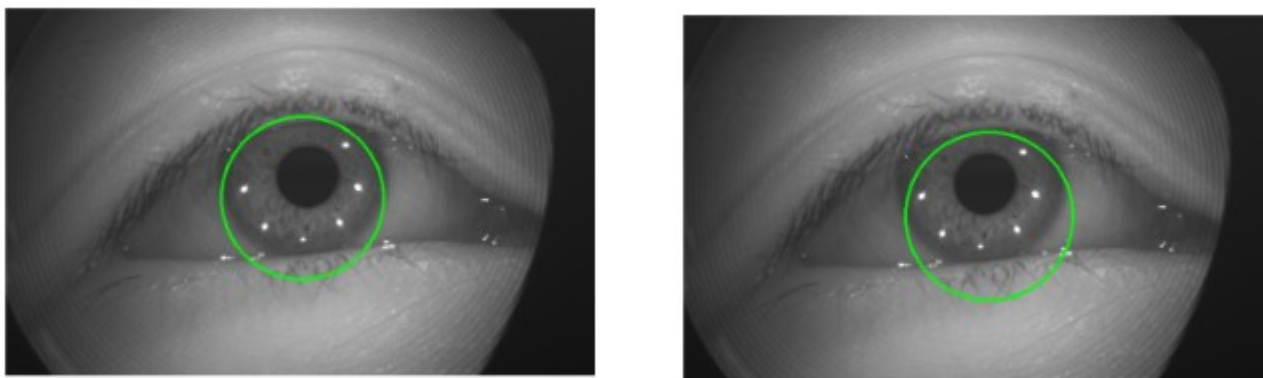


Сл. 10 Споредба на детекција на ирис со MobileNetV2 (лево) и VGG16 (десно)



Сл. 11 Споредба на детекција на ирис со MobileNetV2 (лево) и VGG16 (десно)





Сл. 12 Споредба на детекција на ирис со MobileNetV2 (лево) и VGG16 (десно)

Забележуваме дека ниту еден од моделите нема 100% точност во пронаоѓањето на ирисот. Во случаите каде што окото е подзатворено и ирисот е помалку видлив, VGG16 почесто дава подобри резултати како на сликите 8, 9 и 10. Додека пак MobileNetV2 дава малку подобри резултати во случаите кога ирисот е речиси целосно препознатлив како на сликите 6, 11 и 12.

Бидејќи податочното множество е составено и од слики со полош квалитет, како слики со речиси затворени очи или силна светлина преку камерата којашто го наручува квалитетот на сликата, веројатно е дека таквите слики од податочното множество ја отежнуваат работата на моделите при детекција на ирисот. Бидејќи податочното множество е составено од околу 9.300 слики, би одземало многу време рачно да се отстранат сликите со полош квалитет за можеби да се подобрат перформансите на моделот. Додека пак сликите прикажани овде, се рачно избрани слики коишто моделот ги нема видено претходно при тренирање или тестирање.

## Дескриптори

Дескриптори се начин на претставување на делови од слика коишто имаат уникатни карактеристики коишто може да бидат од корист како нагла промена на боја, појава на рабови или промена на насока на рабовите и сл. Математички се претставуваат како вектори кои содржат броеви кои во себе носат информации за одредени точки од интерес на сликата. Дескрипторите се корисни за споредба на слики бидејќи им овозможуваат на алгоритмите да најдат слични региони и објекти на различни слики.

Во овој проект ќе користиме 3 дескриптори за споредба на сликите. MobileNetV2 од библиотеката Keras и SIFT и ORB дескрипторите од библиотеката OpenCV.

## MobileNetV2 дескриптор

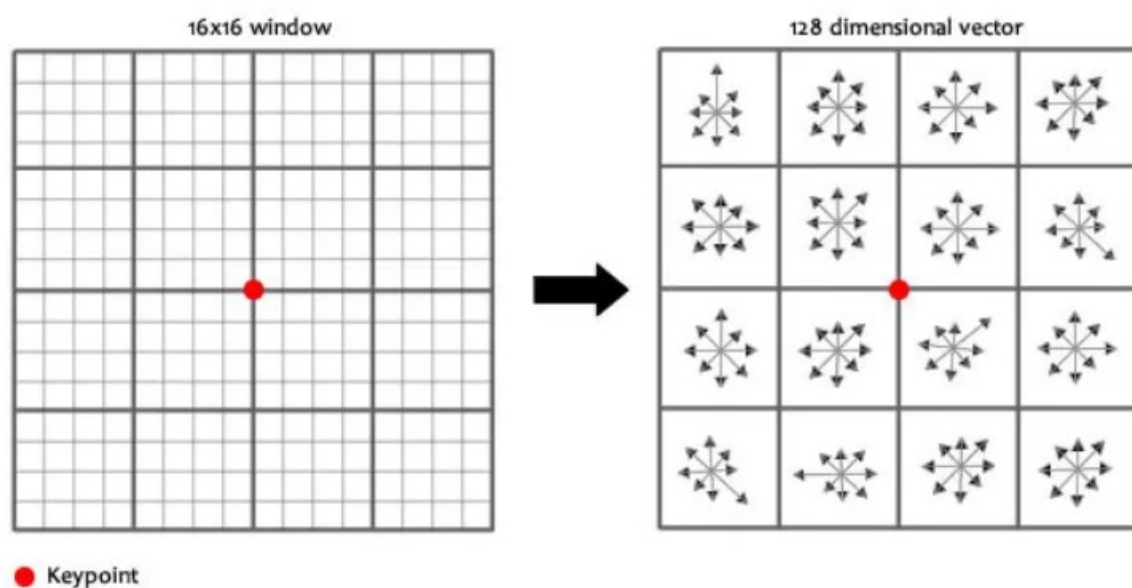
Првиот дескриптор којшто го користиме за споредба на слики го добиваме со користење на преттренираната верзија на MobileNetV2. Во контекст на дескриптори, MobileNetV2 ги

користи конволуциските слоеви на моделот за да открие карактеристични точки и креира нивни вектори што може да се искористат за споредба на две слики.

Ја подготвуваме сликата за обработка така што ја претвораме во 3 канали бидејќи моделот на влез очекува слика во RGB формат и ја претвораме во димензии 224 x 224 како што е потребниот влез на моделот. Потоа ја нормализираме и и додаваме димензија за да симулираме испраќање на batch од слики бидејќи моделот работи со серија од слики на влез дури и ако сакаме да искористиме само една. Исто така користиме потребно претпроцесирање од Keras библиотеката специфично за овој модел пред да го генерираме дескрипторот. На крај дескрипторот го претвораме во една димензија за полесна обработка во следниот чекор.

## SIFT дескриптор

Вториот алгоритам што го користиме е SIFT (Scale-Invariant Feature Transform). Тој е дизајниран за креирање дескриптори коишто нема да се променат при скалирање, ротирање или други трансформации на сликата. Користи Difference of Gaussians метод за да генерира дескриптори за различни скалирања на сликата. На излез добиваме 128 димензионален вектор кој содржи информации за клучните точки и насоката на нивниот градиент.



Сл. 13 SIFT дескриптор на клучна точка

При подготовка на сликата, го подобруваме контрастот за да истакнеме мали детали користејќи ја функцијата CLAHE (Contrast Limited Adaptive Histogram Equalization). Како параметри задаваме лимит на контрастот од 2.0 и блок од 8 x 8 коишто даваат подобри резултати во споредба со други експериментирани вредности. Потоа користиме Canny детектор на рабови со долна граница 100 и горна граница 200. Овие два чекори ги користиме за да ни помогнат при откривањето на клучни точки за дескрипторот. Следно го

иницијализираме SIFT детектор од OpenCV библиотеката со 5000 карактеристики, граница на контраст 0.03, граница на раб 15 и 64 OctaveLayers. Користејќи го овој детектор ги детектираме клучните точки од сликата и генерираме дескриптори. Дескрипторот го претвора во еднодимензионален и потоа доколку ја надминува претходно дефинираната максимална должина од 128, го скратуваме, а доколку е помал од тоа му додаваме нули на крајот. Овој чекор е потребен за дескрипторите на различните слики да имаат иста должина за да може да направиме споредба.

## ORB дескриптор

Последниот дескриптор што го користиме е ORB (Oriented FAST and Rotated BRIEF) дескрипторот којшто е ефикасна алтернатива на SIFT дескрипторот. ORB ги комбинира FAST детекторот на клучни точки и дескрипторот BRIEF. Прво ги пронаоѓа клучните точки користејќи го детекторот FAST и ја пресметува насоката на клучната точка за да овозможи отпорност на ротации. Потоа го користи BRIEF дескрипторот за да генерира бинарен дескриптор којшто е брз за пресметување. За разлика од SIFT и MobileNetV2 дескрипторите, пресметките во ORB дескрипторот се многу побрзи бидејќи на крај добиваме бинарен дескриптор.

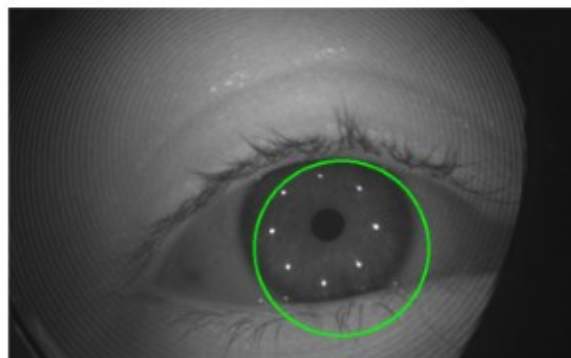
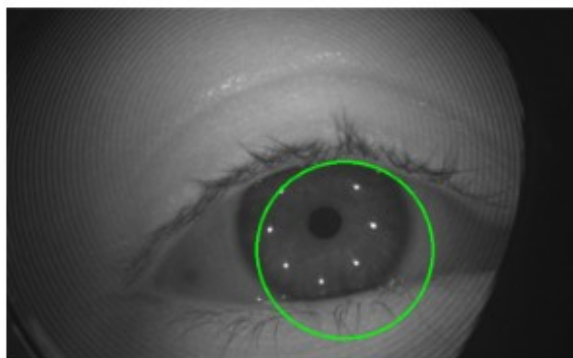
За креирање на дескрипторите, иницијализираме ORB детектор со 5000 карактеристики и потоа ги детектираме клучните точки и генерираме дескриптори за нив. Слично како за SIFT дескрипторот, и овде користиме максимална должина на дескрипторот, но со должина од 1000. Дескрипторот го претвора во еднодимензионален пред да го искористиме во следниот чекор.

## Споредба на дескриптори

На тренираниот модел му даваме две слики коишто претходно ги нема видено и ги користиме добиените резултати за предвидената област на ирисот. Ги користиме трите алгоритми за генерирање на дескриптори коишто беа опишани претходно за да добиеме 3 вида на дескриптори за секоја слика посебно.

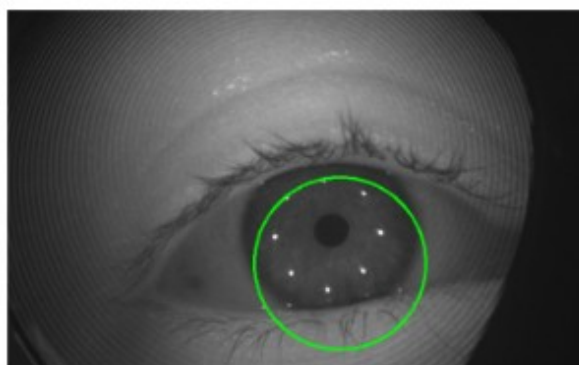
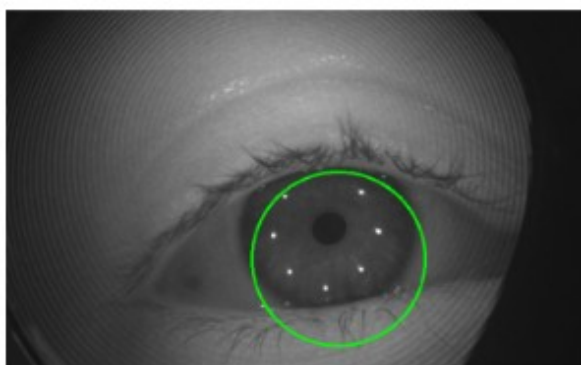
За да споредиме сличност помеѓу ирисите ќе пресметаме сличност помеѓу соодветните дескриптори. За MobileNetV2 и SIFT дескрипторите коишто се покомплексни типови користиме косинусна сличност која пресметува колку се слични насоките на векторите, додека за ORB дескрипторот којшто е составен од бинарни вредности користиме Hamming растојание кое пресметува колку од битовите помеѓу двата дескриптори се различни.

За MobileNetV2 и SIFT, колку пресметаната сличност е поблиска до 1 толку алгоритмот смета дека сликите се послични, додека за ORB, кога пресметаната сличност е поблиска до 0 алгоритмот смета дека сликите се послични.



MobileNetV2  
MobileNetV2 Iris similarity score: 0.9089635610580444  
SIFT Iris similarity score: 0.8302066326141357  
ORB Iris similarity score: 0.018000000000000016

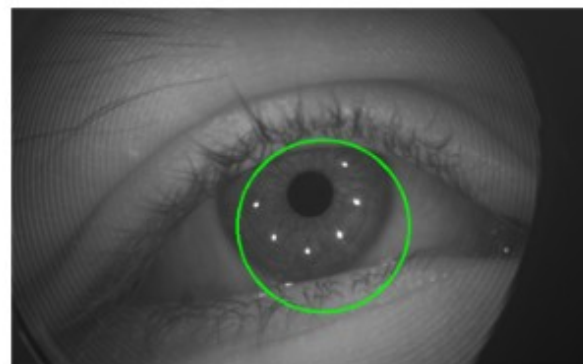
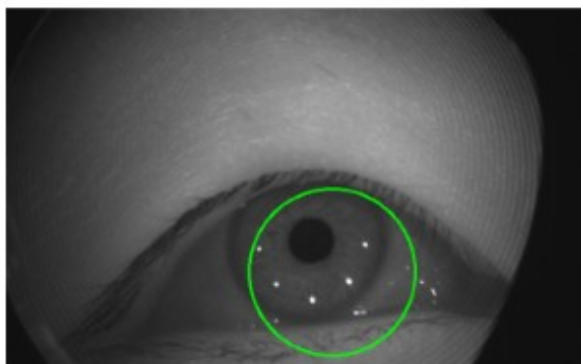
Сл. 14 Генерирање на дескриптори кога детекцијата на ирисот е извршена со MobileNetV2



VGG16  
MobileNetV2 Iris similarity score: 0.9401249885559082  
SIFT Iris similarity score: 0.8470170497894287  
ORB Iris similarity score: 0.236

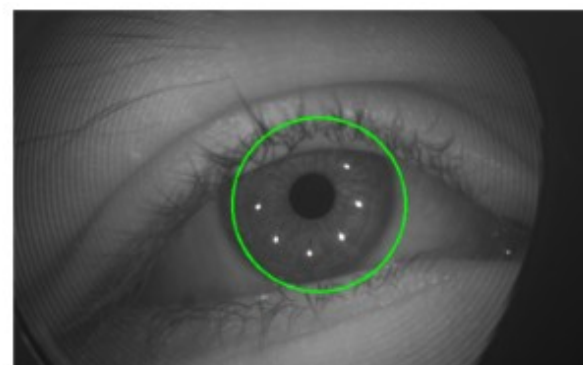
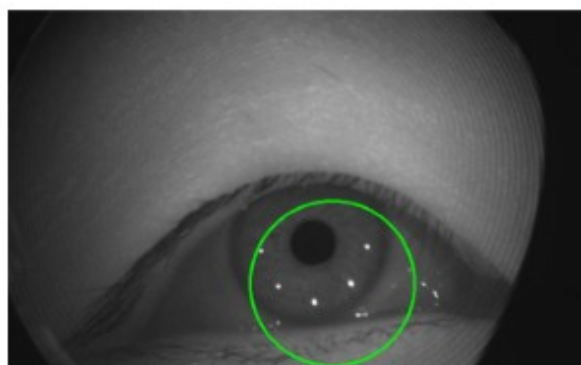
Сл. 15 Генерирање на дескриптори кога детекцијата на ирисот е извршена со VGG16

За сликите 14 и 15 забележуваме дека иако детекцијата на ирисот со VGG16 моделот е минимално полоша за сликата десно, сепак дескрипторите MobileNetV2 и SIFT даваат подобри резултати за тоа дека сликите се многу слични една со друга. Резултатот веројатно се должи на тоа што на сликите е иста личност, па кожата и трепките околу очите се идентични во двете слики. Иако ирисот не е најдобро детектиран, останатиот дел се совпаѓа.



MobileNetV2  
 MobileNetV2 Iris similarity score: 0.6553877592086792  
 SIFT Iris similarity score: 0.8602232933044434  
 ORB Iris similarity score: 0.028000000000000025

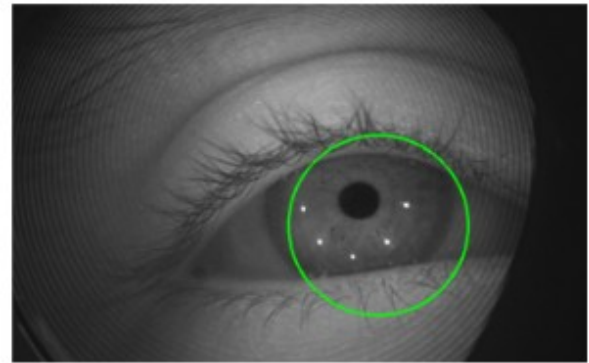
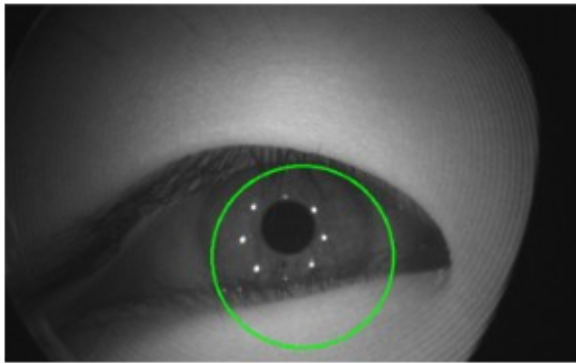
Сл. 16 Генерирање на дескриптори кога детекцијата на ирисот е извршена со MobileNetV2



VGG16  
 MobileNetV2 Iris similarity score: 0.4865327477455139  
 SIFT Iris similarity score: 0.8690400123596191  
 ORB Iris similarity score: 0.021000000000000002

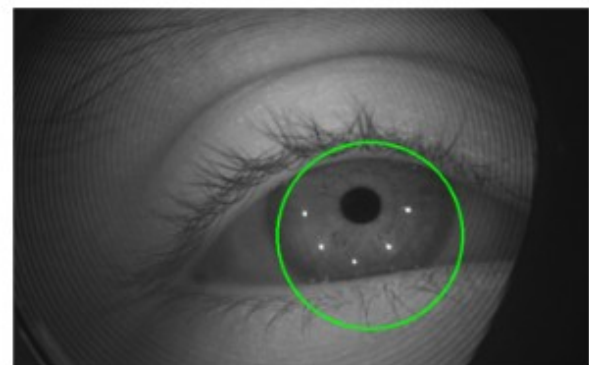
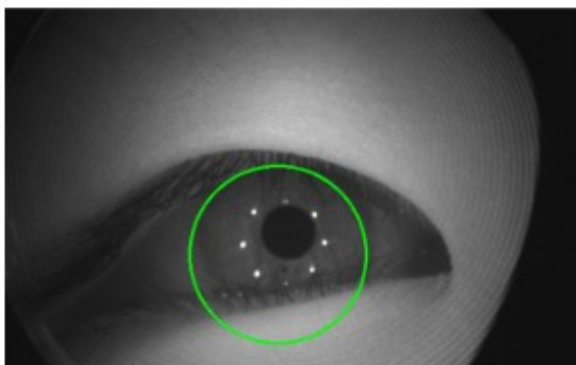
Сл. 17 Генерирање на дескриптори кога детекцијата на ирисот е извршена со VGG16

За сликите 16 и 17 забележуваме дека нема голема промена во резултатите од SIFT и ORB дескрипторите, но резултатот за MobileNetV2 дескрипторот е подобар за слика 17, кога детекцијата на ирисот повеќе се совпаѓа со вистинската положба на ирисот.



MobileNetV2  
MobileNetV2 Iris similarity score: 0.6216187477111816  
SIFT Iris similarity score: 0.7159900665283203  
ORB Iris similarity score: 0.010000000000000009

Сл. 18 Генерирање на дескриптори кога детекцијата на ирисот е извршена со MobileNetV2

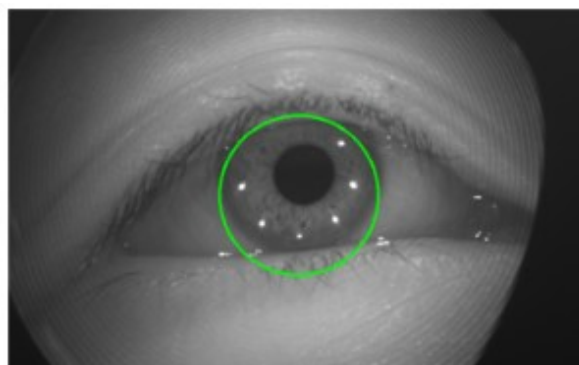
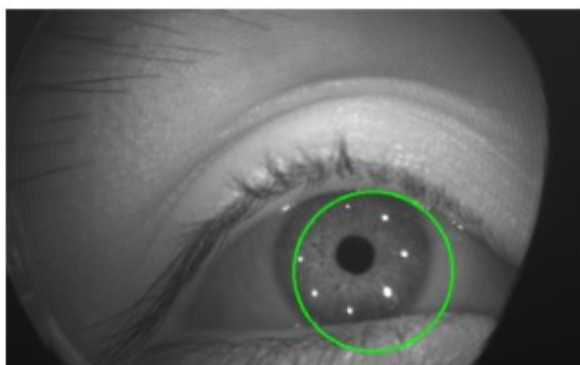


VGG16  
MobileNetV2 Iris similarity score: 0.6315516829490662  
SIFT Iris similarity score: 0.7910184264183044  
ORB Iris similarity score: 0.010000000000000009

Сл. 19 Генерирање на дескриптори кога детекцијата на ирисот е извршена со VGG16

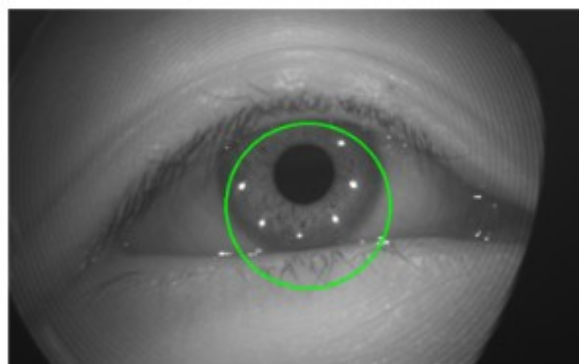
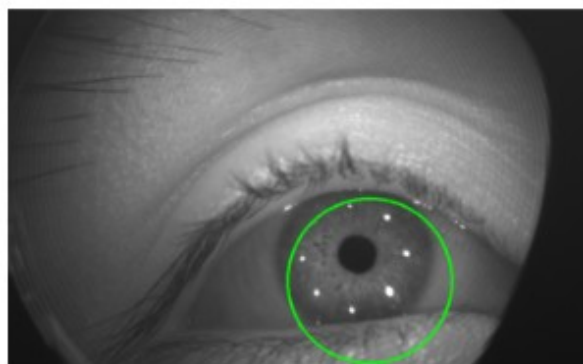
За оваа споредба повторно најдобри резултати добиваме со дескрипторот MobileNetV2. Но, за разлика од претходно, не гледаме голема разлика во резултатите за MobileNetV2, а има промена во резултатите за SIFT дескрипторот. Во овој случај со детекција на ирисот со VGG16 моделот имаме полоши резултати за SIFT иако ирисот е детектиран подобро во споредба со другиот модел.





MobileNetV2  
MobileNetV2 Iris similarity score: 0.46742814779281616  
SIFT Iris similarity score: 0.8191589117050171  
ORB Iris similarity score: 0.045000000000000004

Сл. 20 Генерирање на дескриптори кога детекцијата на ирисот е извршена со MobileNetV2



VGG16  
MobileNetV2 Iris similarity score: 0.46698427200317383  
SIFT Iris similarity score: 0.7748371958732605  
ORB Iris similarity score: 0.0290000000000000026

Сл. 21 Генерирање на дескриптори кога детекцијата на ирисот е извршена со VGG16

Во оваа споредба нема значителна разлика помеѓу двата модели.



## Користена литература

Sandler, Mark, et al. "Mobilenetv2: Inverted residuals and linear bottlenecks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

Yong, Liying, et al. "Application of MobileNetV2 to waste classification." Plos one 18.3 (2023): e0282336.

Tammina, Srikanth. "Transfer learning using vgg-16 with deep convolutional neural network for classifying images." International Journal of Scientific and Research Publications (IJSRP) 9.10 (2019): 143-150.

Vedaldi, Andrea. "An implementation of SIFT detector and descriptor." University of California at Los Angeles 7 (2006).

Rublee, Ethan, et al. "ORB: An efficient alternative to SIFT or SURF." 2011 International conference on computer vision. Ieee, 2011.