

УНИВЕРЗИТЕТ СВ. КИРИЛ И МЕТОДИЈ – СКОПЈЕ

ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО - ФИНКИ



Документација за:

“Визуелни карактеристики за отпечатоци и  
споредба на отпечатоци”

Студент: Милош Кировски – 221058

Ментор: д-р Ивица Димитровски

## **Содржина:**

<u>За проектот.....</u>	<u>3</u>
<u>Преглед на методи.....</u>	<u>4</u>
<u>Типична Итерација со Пример Метод – BRISK.....</u>	<u>9</u>
<u>Статистичка Анализа.....</u>	<u>10</u>
<u>Конволуциски – класификациски модел.....</u>	<u>13</u>
<u>Тестирање на Моделот.....</u>	<u>17</u>
<u>Користена Литература.....</u>	<u>18</u>

# 1. За проектот

## Преглед на проектот

Целта на овој проект е да се истражат и споредат различните методи за споредба на слики со отпечатоци за да се идентификува која техника дава најточни и најефикасни резултати. Конкретно, овој проект се фокусира на евалуација на четири истакнати методи: **SIFT** (Scale-Invariant Feature Transform), **AKAZE** (Забрзана верзија на KAZE), **BRISK** (Бинарни робусни непроменливи скалабилни клучни точки) и **Конволуциска Невронска Мрежа** (CNN). Со анализа на овие методи, ќе ја процениме нивната ефикасност во разликувањето на различни отпечатоци од прсти, земајќи ги предвид факторите како што се робусноста на варијации во обемот, ротацијата и шум.

## Вовед во споредба на отпечатоци од прсти

Споредбата на отпечатоци од прсти е темел на биометриските системи за идентификација, со витална улога во различни домени како што се спроведување на законот, гранична контрола и безбедност на лични уреди. Уникатните шеми на висини и долини на отисовите служат како сигурно средство за идентификација на поединци, имајќи предвид дека нема два слични отпечатоци од прсти, по прецизно, веројатноста за два исти отпечатоци изнесува  $1.5625 * 10^{-11}$  или 1 во 64 милијарди. Оваа вродена уникатност ја прави споредбата на отпечатоците суштинска алатка за проверка на идентитетите.

Сепак, споредбата со отпечатоци од прст не е без свои предизвици. Една од примарните тешкотии лежи во варијабилноста на сликите со отпечатоци од прст. Отпечатоците од прсти може да се фатат под различни услови, што доведува до варијации во обемот, ориентацијата и квалитетот. Дополнително, делумните отпечатоци, бучавата и изобличувањата може дополнително да го комплицираат процесот на споредба. Овие предизвици бараат развој и примена на робусни методи кои можат прецизно и ефикасно да одговараат на отпечатоците од прсти и покрај овие варијации.

## Цел

Примарната цел на оваа документација е да обезбеди сеопфатна споредба на гореспоменатите методи за споредба на отпечатоци од прст. Поточно, имаме за цел:

1. Оценување точност: Со одредување на точноста на секој метод при правилно совпаѓање на сликите со отпечатоци од прст. Ова вклучува проценка колку добро секој метод може да идентификува слични и различни отпечатоци од прсти.
2. Проценување ефикасност: Со споредување на пресметковната ефикасност на секој метод во однос на времето на обработка и користењето на ресурсите.
3. Анализирање робусност: Со испитување како секој метод се справува со варијациите на сликите со отпечатоци, како што се промените во обемот, ротацијата и делумните промени на дадени региони.
4. Давање препораки: Со увид и препораки за соодветноста на секој метод за различни сценарија за препознавање отпечатоци врз основа на експерименталните резултати.

Со постигнување на овие цели, настојувам да обезбедим јасно разбирање за силните страни и ограничувањата на секој метод, помагајќи им н корисниците на наведените методи и истражувачите да донесуваат информирани одлуки при изборот на техники за препознавање отпечатоци.

## 2. Преглед на методи

Со цел обезбедување на темелна споредба, ги искористив следните четири методи за споредба на отпечатоци од прст:

1. **SIFT (Scale-Invariant Feature Transform)**: Развиен од Дејвид Лоу, SIFT е широко користен алгоритам за откривање и опишување локални карактеристики на сликите. Познат е по својата скаларна робусност и отпорност на промени во ротација. SIFT детектира клучни точки и пресметува дескриптори кои се нечувствителни на овие трансформации, што го прави ефикасен за усогласување на сликите со отпечатоци од прст дури и кога има варијации во обемот и ориентацијата на примероците.
2. **AKAZE (Забрзан KAZE)**: AKAZE, подобрување на алгоритмот KAZE, го подобрува оригиналот нудејќи побрзо пресметување додека ја одржува својата скаларна робусност и ротацијата при промени. Користи нелинеарно дифузно филтрирање за откривање карактеристики, што го прави добро прилагоден за ракување со сложени отпечатоци од прсти полни со текстура.
3. **BRISK (Binary Robust Invariant Scalable Keypoints)**: BRISK е уште еден алгоритам за откривање и опис на карактеристики кој се фокусира на бинарни

дескриптори за брзо и ефикасно усогласување. Тој е дизајниран да биде непроменлив за размер и ротација, а неговите бинарни дескриптори ја олеснуваат брзата споредба, што го прави практичен избор за апликации за препознавање отпечатоци во реално време.

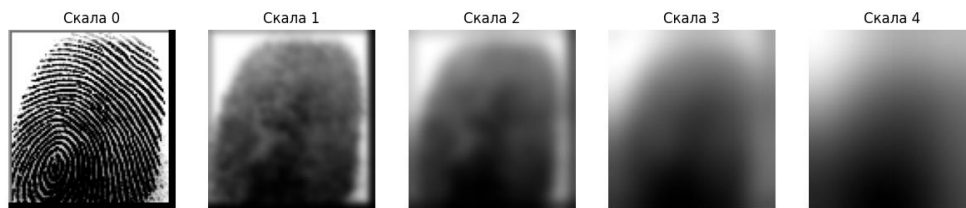
4. **Конволутивна неуронска мрежа (CNN):** Оваа архитектура користи конволуциски слоеви за автоматско извлекување на карактеристики од отпечатоците. Моделот се состои од три конволуциски слоеви со ReLU активација за овозможување на нелинерарност, секој проследени од слоеви за здружување (pooling), и на крај целосно поврзани густе слоеви. Конволутивните слоеви се специјализирани за идентификација на локалните обрасци во сликите, додека целосно поврзаните слоеви ги интегрираат овие карактеристики за класификација на отпечатокот.

Секој метод носи со себе свое множество на предности и ограничувања, а нивните перформанси ќе се оценуваат врз основа на критериуми како што се точност, ефикасност и робусност. Оваа споредба ќе обезбеди вредни сознанија за тоа кои методи се најефикасни за задачите за препознавање отпечатоци.

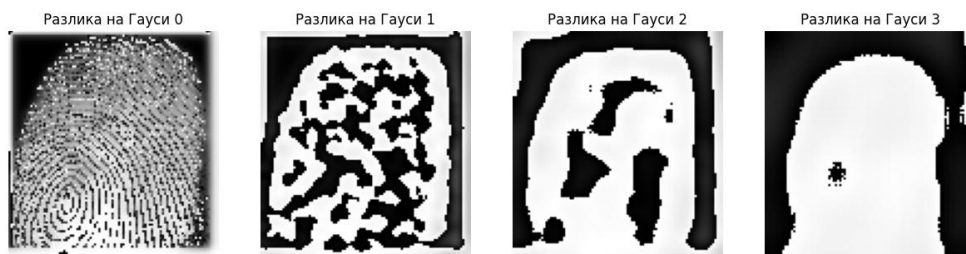
## 1. Како функционира SIFT

- 1.1. **Наоѓање клучни точки низ скали** - Првиот чекор алгоритмот е да се детектираат клучните точки на сликата. Се создава серија слики со различни нивоа на заматување, наречени "простор од скали", каде секоја следна слика има повеќе заматување од претходната. Ова помага да се најдат клучни точки кои се стабилни на различни нивоа на детали. Анализата се прави преку разлика на Гауси (DoG), што се постигнува со одземање на слики од соседни нивоа во просторот од скали.
- 1.2. **Рафинирање на локации на клучни точки** - Откако ќе се идентификуваат потенцијалните клучни точки, SIFT ги рафинира нивните локации. Ова се постигнува со анализа на областа околу секоја клучна точка за прецизно прилагодување на нејзината позиција. SIFT користи математички методи за моделирање на промените во интензитетот на пикселите околу клучната точка. Клучните точки кои се лошо локализирани или имаат низок контраст се отфрлаат, што ја зголемува точноста и робустноста на резултатите.
- 1.3. **Доделување ориентации** - За клучните точки да се неосетливи на ротација, се доделува ориентација на секоја клучна точка врз основа на локалните градиенти. Локалните градиенти се промени во боја или осветленост околу клучната точка и помагаат да се идентификуваат рабовите. SIFT пресметува како и во која насока се менува интензитетот на сликата во околината на клучната точка. Овие насоки се користат за доделување ориентација на клучната точка, што овозможува да се совпаѓаат клучните точки помеѓу различни слики дури и ако се ротирани. Може да се доделат повеќе ориентации на секоја клучна точка за да се покријат различни ротации на околината.

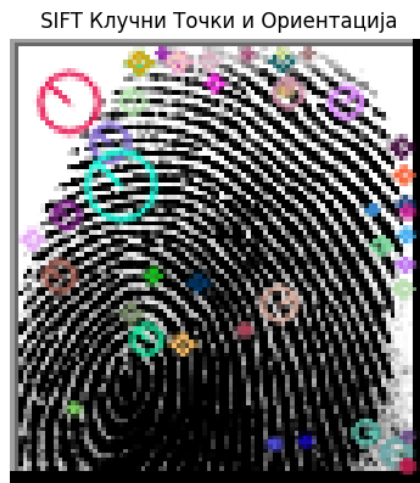
**1.4. Креирање дескриптори** - За да се создаде дескриптор за секоја клучна точка, SIFT дефинира регион од 16x16 пиксели околу клучната точка, кој се дели на 16 помали подрегиони од 4x4 пиксели. За секој подрегион, се пресметуваат големините и насоките на градиентот, што помага да се идентификуваат рабовите. Секој подрегион создава хистограм од 8 кофички за насоки на градиенти, каде што секоја кофичка претставува опсег на насоки. Со комбинирање на хистограмите од сите 16 подрегиони, се формира 128-димензионален дескрипторски вектор кој ја доловува локалната текстура и структура околу клучната точка.



Слика 1: Простор од Скали



Слика 2: Разлика на Гауси



Слика 3: Клучни Точки и Ориентација кај SIFT

## **2. Како функционира AKAZE**

### **2.1 Наоѓање клучни точки низ скали - Алгоритмот започнува со откривање**

клучните точки користејќи нелинеарно дифузно филтрирање. Овој процес вклучува измазнување на сликата на начин кој се прилагодува на карактеристиките на сликата, зачувувајќи ги рабовите и карактеристиките додека го намалува шумот. Методот на Брза Експлицитна Дифузија (FED), го забрзува овој процес со ефикасно применување на нелинеарна дифузија за создавање на повеќе нивоа на размер на сликата. Процесот на дифузија овдека се однесува на техниката на измазнување на сликата на начин што нивото на измазнување варира врз основа на локалните карактеристики на сликата. FED го забрзува овој процес во споредба со традиционалните методи, како Гаусовото заматување, кое применува еднообразно измазнување и е помалку приспособливо на локалните карактеристики.

### **2.2 Рафинирање на локации на клучни точки – По идентификувањето на потенцијалните клучни точки, AKAZE ги рафинира нивните локации со анализа на нивната околина. За да се постигне поголема прецизност, Scharr филтри се користат за подобрување на ротационата непроменливост. Овие филтри се филтри за откривање рабови кои ги нагласуваат промените во градиентот на сликата, ја зголемуваат остријата на рабовите и ја подобруваат точноста на откривање на клучните точки со попрецизно идентификување на рабови и транзиции.**

### **2.3 Доделување ориентации - AKAZE доделува ориентации на клучните точки за да обезбеди додатна ротациона непроменливост. За секоја клучна точка, се пресметуваат локални градиенти и нивната дистрибуција во соседството. Оваа информација ја дефинира главната ориентација на клучната точка, овозможувајќи совпаѓање на клучните точки помеѓу различни слики, дури и ако тие се ротираат.**

### **2.4 Креирање дескриптори - Дескрипторите го користат Modified Local Difference Binary (MLDB) методот за да генерираат компактни и ефикасни бинарни дескриптори за секоја клучна точка. MLDB доловува информации за локалната текстура со енкодирање на релативните разлики во интензитетот на пикселите околу клучната точка во бинарна низа. Овој процес вклучува споредба на парови вредности од пиксели во дефиниран регион, што резултира во бинарен код кој го претставува локалниот изглед на клучната точка. Методот MLDB е дизајниран да биде брз и ефективен, обезбедувајќи робусни описи на клучните точки со ниски пресметковни трошоци.**

Со оглед на ова, може да се забележи дека AKAZE е доста брз и користи доста различен алогоритал од SIFT, но поради самата структура на алгоритмот, AKAZE во нашето истажување ќе се покаже како добар во препознавање на слики од отпечатоци кои не се изменети, но лош во слики со мали промени.

### 3. Како функционира BRISK

#### 3.1 Наоѓање клучни точки – Алгоритмот започнува со наоѓање на клучните точки

користејќи метод за откривање агли. Процесот ги идентификува точките од интерес, како што се аглите или рабовите, кои веројатно ќе бидат карактеристични и стабилни на различни слики. За ова цел го користи алгоритмот AGAST (Adaptive and Generic Accelerated Segment Test). AGAST открива агли преку анализа на варијациите на интензитетот во локалното соседство.

#### 3.2 Усовршување на клучните точки - Откако ќе се идентификуваат потенцијалните

клучни точки, ги рафинира нивните локации за да ја подобри точноста. Ова се постигнува со вклопување на попрецизен модел на локалното соседство околу секоја клучна точка.

#### 3.3 Доделување ориентација - За да се постигне неосетливост на ротацијата, BRISK

доделува ориентација на секоја клучна точка. Ова се прави со пресметување на насоката на доминантниот градиент во локалниот регион на клучната точка. Со порамнување на дескрипторите на клучните точки со оваа доминантна ориентација, алгоритмот гарантира дека дескрипторите остануваат конзистентни дури и ако сликата е подложена на ротација. Овој чекор е клучен за усогласување на клучните точки на слики кои имаат различни ориентации.

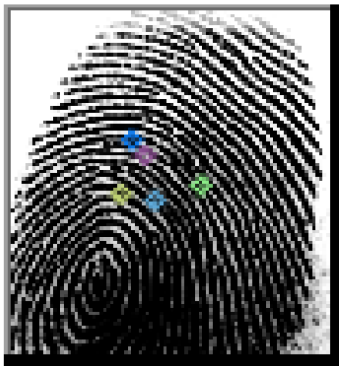
#### 3.4 Екстракција на дескриптор - BRISK генерира бинарни дескриптори за да ја

претстави секоја клучна точка. Дескрипторите се креираат со користење на специфични шеми на примерок од точки околу клучната точка. Алгоритмот ги споредува вредностите на интензитетот на овие точки и ги енкодира резултатите во бинарна низа. Бинарните дескриптори даваат информации за локалната текстура и се отпорни на промени во осветлување и шум.

#### 3.5 Усогласување на дескриптори – На крај, за усогласување на клучни точки помеѓу

различни слики, BRISK користи Хамингтоново растојание за да споредува бинарни дескриптори. Ова растојание го мери бројот на различни битови помеѓу две бинарни низи, обезбедувајќи брз и ефикасен начин за наоѓање совпаѓања.

AKAZE Клучни Точки



Слика 4: AKAZE Клучни Точки

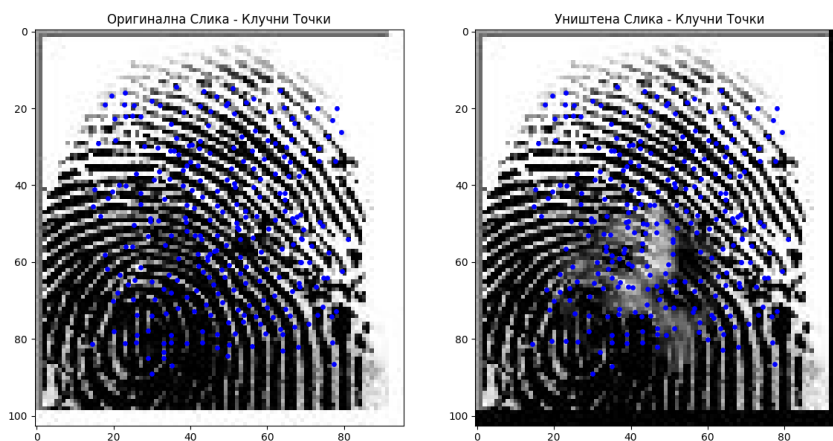
BRISK Клучни Точки



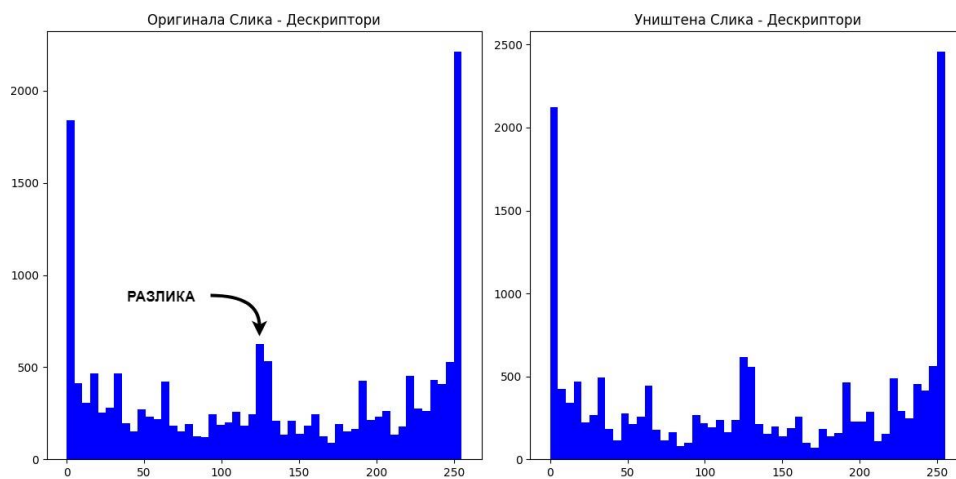
Слика 5: BRISK Клучни точки



### 3. Типична Итерација со Пример Метод – BRISK



Слика 6: Наоѓање Клучни Точки



Слика 7: Претстава на Дескриптори со Хистограм

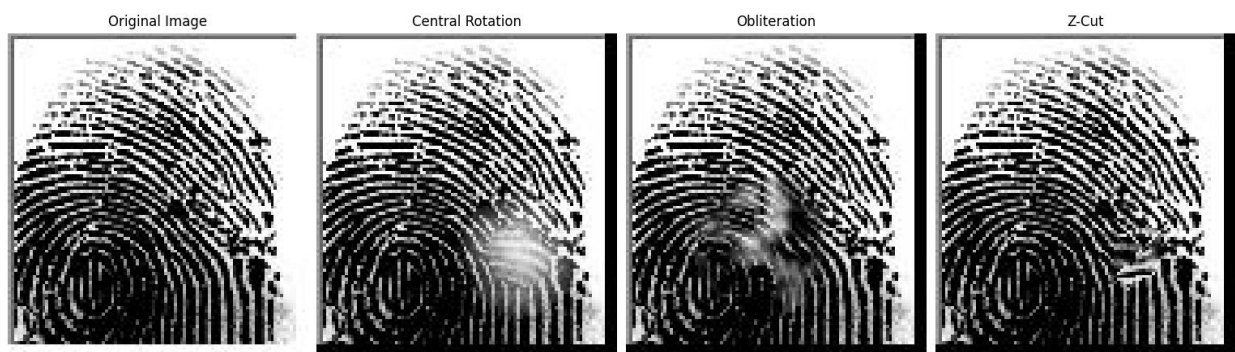


Слика 8: Совпаѓања со Праг при Споредба

## 4. Статистичка анализа

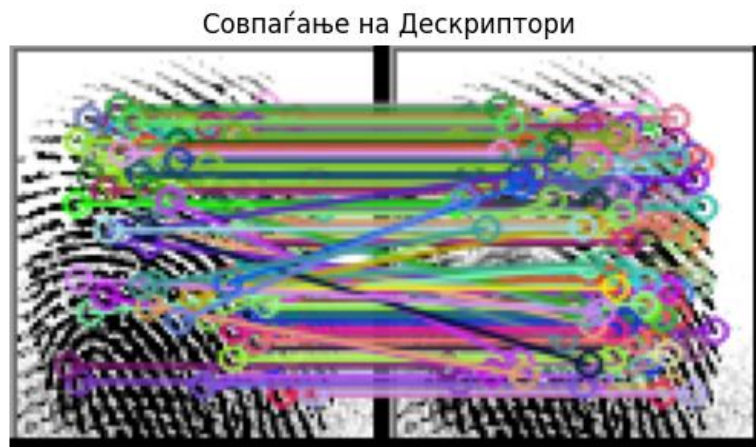
Прво ќе погледнеме како изгледа базата на отпечатоци со кои работиме како и примерок кој опишува како работи овој проект. Отпечатоците се организирани во две множества:

- Правилни отпечатоци: Ова се правилните 10 верзии на отпечатоци. Проектот е граден околу идејата дека при дадена сменета слика од истиот отпечаток, ќе може да ја добиеме правилната верзија на истиот.
- Сменети отпечатоци: Ова се слики исти како правилните отпечатоци со некоја промена која ќе претставува проблем при споредување. Промените доаѓаат во три облици: централна ротација, уништување и сечење на даден дел од сликата.



Слика 6: Примерок од Базата на Отпечатоци

Всушност сега претходно опишаните алгоритми, со најдени клучни точки и дескриптори на двата примероци, може да направат споредба помеѓу две вакви слики и да дадат резултат за тоа дали овие две слики се исти. Истиот резултат може и да се визуелизира за да ги видиме самите поврзувања на клучните точки кои се исти помеѓу двете слики



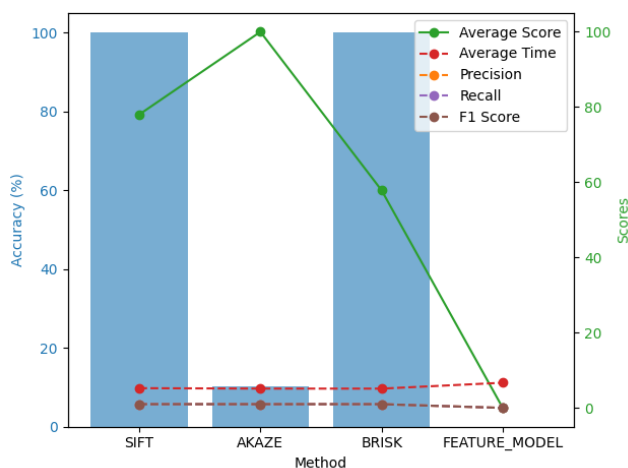
Слика 7: Совпаѓање на SIFT дескриптори

## Преглед на анализа:

Во следната анализа, ги оценуваме трите методи за совпаѓање на отпечатоци од прст - SIFT, AKAZE и BRISK - за да ја одредиме нивната ефикасност во идентификувањето на изменетите отпечатоци од прсти. Ја извршивме истата скрипта низ збир на изменети и реални слики, изведувајќи перформансни метрики кои потоа ги анализиравме за да ја процениме веродостојноста и ефикасноста на секој метод.

	Метод	Точност	Точни Совпаѓања	Вкупно Слики	Просечен Резултат	Просечно Време
0	SIFT	100.0	29	29	78.01908	5.24959
1	AKAZE	10.345	3	29	100.0	5.13784
2	BRISK	100.0	29	29	57.86181	5.12799

Слика 8: Табела за Перформанси на Методите



Слика 9: Евалуациски Метрики по Метод

## Точност

Точноста на методот ја одразува неговата способност правилно да ги идентификува изменетите отпечатоци од прсти. Анализа откри дека и SIFT и BRISK постигнаа совршена точност, успешно идентификувајќи ги сите видови изменети слики. Ова покажува дека овие методи се многу сигурни кај задачите за идентификација на отпечатоци во овој контекст. Спротивно на тоа, AKAZE покажа значително помала точност. И покрај добрите резултати во другите метрики, ниската прецизност на AKAZE сугерира дека истиот има проблем со правилно идентификување на изменетите отпечатоци од прсти, ограничувајќи ја нејзината ефикасност во оваа специфична апликација. AKAZE притоа добро се покажува при барање на копија на дадена слика за разлика од сменета верзија.

## **Просечен резултат**

Просечниот резултат е мерка за квалитетот на совпаѓањето доделен од секој метод при споредба на дескриптори и клучни точки. AKAZE постигна совршен просечен резултат, но тоа не е во корелација со неговите перформанси во однос на точноста. Ова несовпаѓање покажува дека иако AKAZE може да обезбеди високи оценки за квалитет, тоа не мора да значи дека работи добро во практични сценарија за совпаѓање. Од друга страна, SIFT и BRISK покажаа пониски просечни резултати, што се усогласува со нивните набљудувани перформанси. Просечниот резултат дава увид во квалитетот на произведените совпаѓања, но мора да се толкува заедно со точноста за да се разбере севкупната ефективност.

## **Просечно време**

Просечното време на извршување го претставува времето потребно на секој метод за обработка и совпаѓање на сликите. Разликите во времето на извршување меѓу методите се минимални, што сугерира дека временската ефикасност не е значајна разлика во овој контекст. Но ова не значи дека ова важи генерално. Секој од методите различно одговара на текстура, контури и други шеми низ сликата. Во нашиот случај, за мало-димензионални слики од отпечатоци, разликите се незабележливи.

## **Најдобар метод за овој проект**

Нашата детална анализа нагласува дека SIFT и BRISK се многу ефикасни за усогласување на отпечатоци од прсти со нивни алтернативни, изменети верзии, постигнувајќи совршена точност. Овие методи се сигурни за прецизно идентификување на изменетите отпечатоци од прсти и обезбедување постојан квалитет со многу голем број на клучни точки. AKAZE, и покрај високиот просечен резултат, паѓа при метриката за прецизност и практични перформанси, што го прави помалку погоден за специфичната намена на овој проект. Минималните разлики во времето на извршување сугерираат дека временската ефикасност не е значаен фактор во овој проект. Генерално, SIFT и BRISK се најпосакуваните методи засновани на нивните супериорни перформанси и доверливост во задачите за идентификација на отпечатоци од прст.

Во заклучокот, важно е да се напомене дека овие методите нудат одлични резултати за идентификација на отпечатоци, постојат и други современи технологии. Конволуционите невронски мрежи (CNN) се еден од најсовремените пристапи во обработката на слики и препознавање на образци. Затоа, последниот пристап ќе ни биде точно ваков модел.

## 5. Конволуциски – класификациски модел

Во овој дел ќе го разгледаме четвртиот пристап, кој користи конволуциски модел, имплементиран со користење на TensorFlow и OpenCV. Моделот е специјално дизајниран за препознавање и класификација на слики од отпечатоци од прст, и ја користи архитектурата на конволуциска невронска мрежа (CNN). Во оваа секција ќе ги објаснам клучните компоненти и дизајнерските избори зад моделот, со фокус на образложение зад секоја одлука при градење на моделот.

**Потребните библиотеки** - Моделот започнува со импортирање на основни библиотеки:

- **NumPy**: Се користи за операции со низи и нумерички пресметки, кои се основни за обработка на слики и манипулација со податоци.
- **OpenCV**: Се користи за читање и преработка на сликите од отпечатоци, клучна библиотека која служи да се осигураме дека сликите се во правилен формат и големина за моделот.
- **TensorFlow**: Основната рамка за длабоко учење што се користи за градење, обука и оценување на CNN моделот.
- **scikit-learn**: Обезбедува алатки за поделба на базата на податоци на множеста за тренирање и тестирање, како и обезбедување на опширни информации за тренинг процесот и крајните перформансите на моделот.

### Зголемување на сликата

Функцијата **augment\_image** е од витално значење за подобрување на разновидноста на базата на примероци, токму поради самиот факт што моделот е градан над мал податочна база со мал број на слики. Зголемување на бројот на примероци се применува за да се спречи префитување со вештачко зголемување на големината на множеството за тренирање преку различни трансформации. Поточно, оваа функција го користи ImageDataGenerator на TensorFlow за да примени:

- **Опсег на ротација**: ротирање во опсег од 10 степени за симулирање на различни ориентации на отпечатокот од прст.
- **Поместувања во ширина и висина**: Мали поместувања (10%) во двете насоки за да имитираат варијации во поставувањето отпечатоци од прст.
- **Опсег на зумирање**: Опсег на зумирање од 10% за да се земат во предвид малите разлики во скалирањето на отпечатоците за време на снимањето.

Овие зголемувања му помагаат на моделот да генерализира подобро со тоа што го учи да препознава отпечатоци од прсти и покрај различните мали варијации во ориентацијата, положбата и размерот.

## Вчитување и преобработка на слики

Функцијата **load\_images\_from\_folder** е дизајнирана да вчита слики од отпечатоци во сиви тонови (grayscale), да ја промени големината на 224x224 пиксели и да ги претвори истите во формат соодветен за CNN. Изборот за промена на големината на сликите на 224x224 пиксели се усогласува со вообичаените големини влез за CNN, но и ни овозможува да имаме униформен и единечен влез, независно од големините на сликите во нашата база.

Со редување на сликите со сиви тонови, над сите три канали, создаваме RGB слика што одговара на очекуваниот влезен облик за конволуционите слоеви, иако оригиналната слика не е RGB. Оваа трансформација обезбедува компатибилност со архитектурата на ваквата мрежа без да се менува основната содржина на сликата.

Нормализацијата на вредностите на пикселите во опсегот  $[0, 1]$  се изведува за да се подобри стабилноста и брзината на процесот на тренирање. Овој чекор на нормализација е клучен за спречување на проблеми како што се исчезнување или експлозија на градиентите за време на тренирање на моделот.

## Подготовка на збирката на податоци

Функцијата **prepare\_data** ја генерира конечната база на податоци за обука со зголемување на секоја слика и доделување етикети. Етикетите се one-hot енкодирани, што е формат кој го бара функцијата за загуба **categorical\_crossentropy**. one-hot енкодирање ги трансформира ознаките на класите во формат на бинарна матрица, каде што секоја класа е претставена како бинарен вектор.

Овој метод му овозможува на моделот да калкулира веројатности за секоја класа, како и ефективно да се справи со задачите за класификација со повеќе класи. Екстензивната употреба на зголемување на податоци осигурува дека моделот се тренира на разновидна и репрезентативна база на податоци, што е од клучно значење за постигнување добра генерализација на новите слики со отпечатоци од прсти.

## Создавање модел

Срцето на овој модел лежи во самата архитектура, дефинирана во функцијата `create_simple_model`. Моделот е дизајниран со следните слоеви:

- **Conv2D Сроеви:** Мрежата вклучува три конволуциски слоеви, секој со ReLU (Поправена линеарна единица) активирање. ReLU е избрана како функција за активирање бидејќи воведува нелинеарност додека е пресметковно ефикасна, помагајќи ѝ на мрежата да научи сложени шеми во сликите. Зголемениот број на филтри (32, 64, 64) како што се продлабочува мрежата му овозможува на моделот да долови покомплексни детали на повисоки нивоа.
- **MaxPooling2D слоеви:** Сроевите за здружување го следат секој конволуциски слој за да ги намалат мапите на карактеристики, намалувајќи ги просторните димензии, но и бројот на параметри во моделот. Овои слоеви му помагаат на моделот да се спречи со префитување преку фокусирање на најважните карактеристики и го намалуваат пресметковниот товар.
- **Flatten Слој:** Овој слој ги израмнува 2D мапите на карактеристики во 1D вектор, што е неопходно пред да се внесат податоците во целосно поврзаните слоеви. Го поврзува јазот помеѓу конволуциските слоеви (кои ги учат просторните хиерархии) и густите слоеви (кои ги прават конечните предвидувања).
- **Густи Сроеви:** Моделот има еден целосно поврзан слој со 64 јазли и ReLU активација, проследен со излезен слој со `num_classes` јазли и функција за активирање **softmax**. ReLU активирањето во густиот слој гарантира дека мрежата може да научи сложени комбинации од карактеристиките извлечени преку конволуциските слоеви. Излезниот слој го користи softmax активирањето за да ги конвертира конечните излези во веројатносна распределба, каде што секоја вредност ја претставува веројатноста сликата да припаѓа на одредена класа.

Изборот на еден густ слој по израмнувањето е намерен; бидејќи воспоставува рамнотежа помеѓу комплексноста на моделот и перформансите. Додавањето повеќе слоеви може да доведе до префитување, особено со оглед на релативно малата големина на множеството слики со кои работиме, додека еден густ слој обезбедува сосема доволен капацитет за учење од карактеристиките извлечени преку конволуционите слоеви.

## Компилација и обука на модел

Моделот е компајлиран со помош на оптимизатор **Adam**, кој е добро прилагоден за обука на длабоки невронски мрежи поради неговата можност за адаптивна стапка на учење. Adam ги комбинира предностите на два популарни методи за оптимизација - **AdaGrad** и

**RMSProp** - што го прави робустен за справување со ретки градиенти при проблеми со шум како што е класификација на слики.

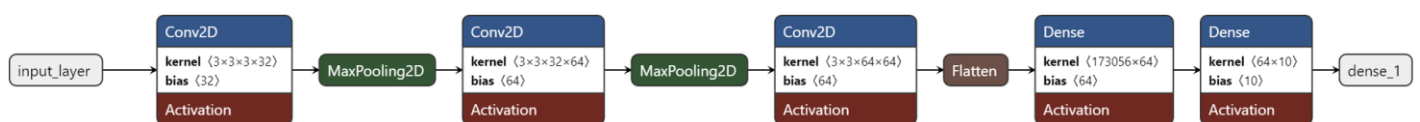
Функцијата за загуба **categorical\_crossentropy** е избрана затоа што проблемот што се работи е проблем на класификација со повеќе класи. **categorical\_crossentropy** ја мери разликата помеѓу предвидената веројатносна дистрибуција и вистинската дистрибуција (one-hot енкодирани етикети), обезбедувајќи јасна метрика која моделот ја минимизира за време на тренирањето.

Тренирањето се изведува во текот на 10 епохи со големина на пакет од 32 слики, што е стандарден избор кој ја балансира брзината на тренирање со стабилноста на конвергенцијата. Употребата на валидациско поделување овозможува моделот да се оценува на подмножество од податоците за тренирање по секоја епоха.

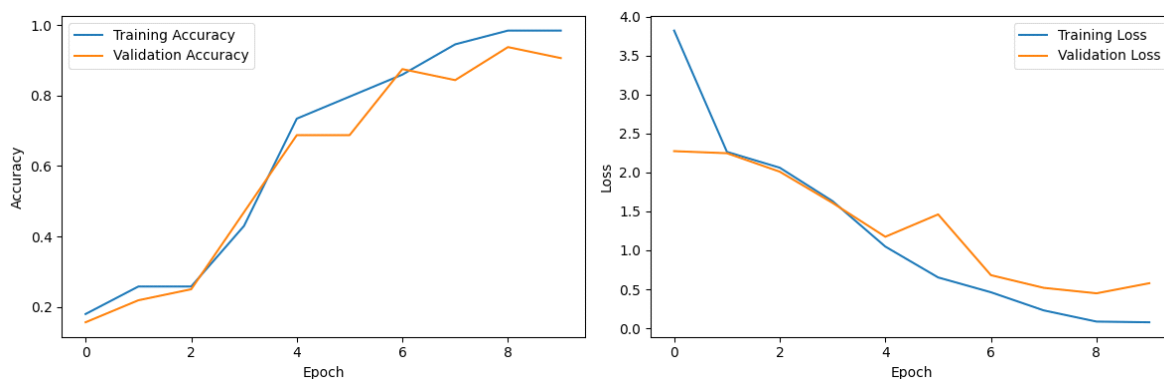
### Евалуација и зачувување на моделот

По тренирањето, моделот се оценува на тест сет за да се одреди неговата точност, обезбедувајќи квантитативна мерка за неговите перформанси. Во нашиот случај, моделот постигна 99,83% стапка на точност по 10 епохи.

Конечно, обучениот модел е зачуван во **HDF5 формат**, кој ги чува архитектурата, тежините и тренинг конфигурацијата. Ова овозможува лесно пренесување и вчитување како и дополнително доработување на моделот или евалуација без потреба од тренирање на моделот од нула.



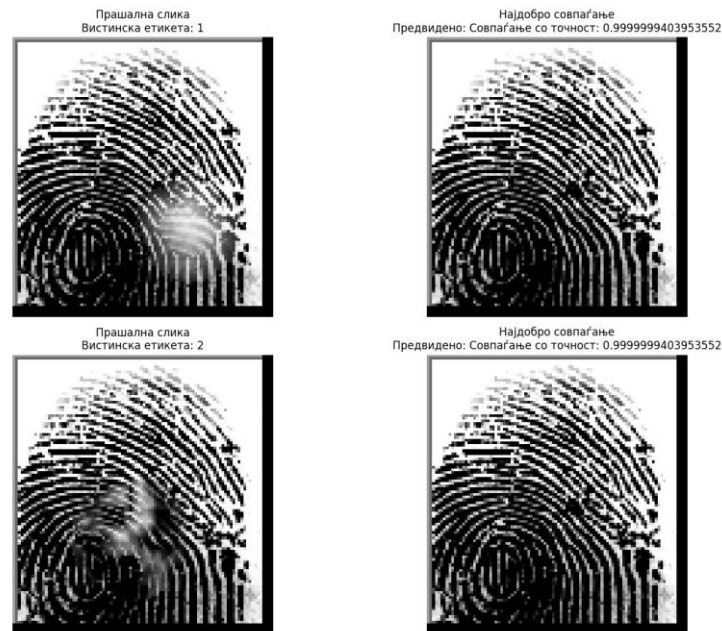
Слика 10: Архитектура на Моделот



Слика 11: Точност и Губиток при Фаза на Тренирање

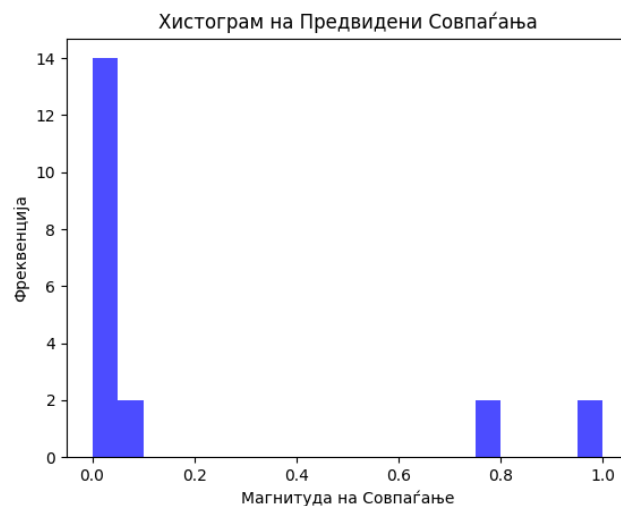


## 6. Тестирање на Моделот



Слика 12: Наоѓање на Совпаѓања при Прашална Слика со CNN Модел

Всушност лесно може да добиеме увид во тоа колку мисли моделот дека секоја слика од базата на отпечатоци наликува на прашалната сменета слика. Во следниот хистограм кој ја прикажува распределеноста на одлуките при преглед на целата база отпечатоци, ја користиме истата слика како и погоре.



Слика 13: Хистограм на Предвидувања

## 7. Користена Литература

1. Lowe, D. G. (2004). "Distinctive Image Features from Scale-Invariant Keypoints." International Journal of Computer Vision – [Линк до труд](#)
2. BRISK: Binary Robust Invariant Scalable Keypoints ETH Zurich – [Линк до труд](#)
3. Accelerated Embedded AKAZE Feature Detection Algorithm on FPGA – [Линк до труд](#)
4. Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*. Springer – [Линк](#)
5. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. - [Линк](#)
6. Khan Academy. (2018, October 10). *Introduction to Convolutional Neural Networks (CNNs)* [Video]. YouTube. [https://www.youtube.com/watch?v=YRhxdVk\\_sIs](https://www.youtube.com/watch?v=YRhxdVk_sIs)
7. NeuralNine. (2022, February 20). *Fingerprint Matching in Python* [Video]. YouTube. <https://www.youtube.com/watch?v=IIvfqfKkiio>