

Факултет за информатички науки и компјутерско инженерство
Универзитет "Св. Кирил и Методиј" - Скопје

Примена на облачиња и пресметување со магла

Докторска дисертација

м-р Александар Бахтовски

Ментор:
Проф. д-р Марјан Гушев

Работна верзија
Скопје, Октомври 2017

Application of cloudlets and fog computing

PhD thesis

MSc Aleksandar Bahtovski

Резиме

Мобилните уреди се важен дел од секојдневието на луѓето и претставуваат најважна и најсоодветна комуникациска алатка, овозможувајќи им различни функционалности. Бројот на нови апликации наменети за потребите на корисниците на мобилни уреди, секојдневно се зголемува. Заради ограниченоста на сопствените ресурси, мобилните уреди имаат потреба од користење на дополнителни ресурси. Алтернативна метода е користење на надворешни процесирачки единици, кои им дозволуваат на крајните корисници да ги користат нивните ресурси за извршување на мобилните апликации.

Тековните методологии за развој на мобилни апликации кои користат надворешни ресурси за извршување (облак, облаче, магла, роса) сеуште не се усовершени до таа мера да целосно се користат во развојот на денешните мобилни апликации. Постојат голем број на недостатоци и нејаснотии кои се главна причина што сеуште имплементацијата на ваквите методологии е во рана фаза.

Постојат различни архитектурни решенија за забрзување на процесирањето на мобилните уреди кои вклучуваат префрлување на пресметувањето на сервер, како што се, пресметување во облак, облаче, магла или роса.

Главна цел на докторската дисертација е да моделира однесување на мобилна апликација којашто користи алгоритам за оптичко препознавање знаци, зборови и фрази, со користење на различни архитектурни решенија за пресметување во облак, облаче, магла или роса и да даде одговор кое решение е оптимално.

Преку низа теоретски и експериментални истражувања, како и преку моделирање на перформансите на различните архитектурни решенија, се одговори на прашањето за најдобар архитектонски пристап - облачето. За практична примена на заклучоците од истражувањето, беше развиен и соодветен бизнис модел.

Abstract

Mobile devices are an important part of people's everyday life and represent the most important and relevant communication tool, providing them with different functionalities. The number of new applications designed for the needs of mobile users is growing every day. Due to the limitations of their own resources, mobile devices need to use additional resources. An alternative method is to use external processing units, which allow end users to use their resources to run mobile applications.

The current methodologies for the development of mobile applications that use external resources for execution (cloud, cloudlet, fog computing, dew computing) are not yet perfected to the full extent used in the development of today's mobile applications. There are a large number of shortcomings and uncertainties which are the main reason that the implementation of such methodologies is still at an early stage.

There are various architectural solutions to speed up the processing of mobile devices that involve offloading to server computing, such as, cloud computing, cloudlet computing, fog computing or dew computing.

The main goal of the doctoral dissertation is to model the behavior of a mobile application using an algorithm for optical character recognition of words and phrases, using a variety of architectural solutions for computing in cloud, cloudlet, fog or dew, and define an optimal solution.

Through a series of theoretical and experimental research, as well as through modeling the performance of various architectural solutions, the answer to the question of the best architectural approach is the cloudlet. For practical implementation of the conclusions of this research, an appropriate business model was developed.

Содржина

1	Вовед	11
1.1	Мотивација	12
1.2	Предмет на истражувањето	13
1.3	Цели на истражувањето	14
1.4	Придонес од истражувањето	15
2	Основни концепти	17
2.1	OCR (Оптичко препознавање на карактери)	17
2.2	Пресметување во облаче	17
2.3	МОСНА архитектура	18
2.4	Пресметување во магла (Fog computing)	19
2.5	Пресметување во роса (Dew computing)	21
2.6	Бизнис модел на решението	23
2.7	Споредба на архитектурните пристапи на OCR алгоритмот со мобилно пресметување	24
3	Преглед на тековни истражувања	25
3.1	Примена на OCR кај мобилни уреди	25
3.2	Архитектурни пристапи	25
3.2.1	Пресметување во облаче	26
3.2.2	Пресметување во магла	26
3.2.3	Пресметување во роса	27
3.3	Истражувања од областа на темата	27
4	Решенија за OCR со мобилен уред	31
4.1	OCR процесирање	31
4.1.1	Архитектура на Tesseract	31
4.1.2	Оптичко скенирање	32
4.1.3	Детектирање на косина и прет-процесирање	32
4.1.4	Бинарно прет-процесирање	32
4.1.5	Сегментација	33
4.1.6	Извлекување на карактери	34
4.1.7	Препознавање на карактерите и постпроцесирање	35

4.2	Различни пристапи за процесирање	35
4.2.1	<i>Сценарио M</i>	36
4.2.2	<i>Сценарио L</i>	37
4.2.3	<i>Сценарио F</i>	37
4.2.4	<i>Сценарио C</i>	37
4.3	Опис на решението	38
5	Методологија за експериментално истражување	41
5.1	Тестна околина	41
5.2	Прилагодувања за експериментот	42
5.3	Тестни сценарија	43
5.4	Тестни податоци	45
5.5	Цел на експерименталното истражување	46
6	Евалуација на експерименталните резултати	47
6.1	Перформанси за процесирање	47
6.2	Ефектот на OCR процесирањето	48
6.3	Зависност од комплексноста на сликата	48
6.4	Точност	54
6.5	Зависност од обемот на податоци за обработка	55
6.6	Анализа на потрошувачката на батерија	57
6.7	Анализа на добиените резултати	58
7	Моделирање на однесувањето на времето на процесирање	61
7.1	Моделирање на зависноста од бројот на карактери	62
7.2	Моделирање на зависноста од големината на сликата	62
7.3	Анализа на најдоброто решение	66
7.4	Споредба со други пристапи	67
8	Моделирање на перформанси за различни архитектурни решенија	71
8.1	Анализа на перформансите	71
8.2	Задоцнување во комуникацијата	72
8.3	Анализа на зависноста од бројот на процесирани карактери	74
8.4	Вкупно време на процесирање	76
8.5	Отстапувања помеѓу моделот и експерименталните резултати	79
8.6	Заклучоци од теоретскиот пристап за разгледување на перформансите	80
9	Дискусија и заклучни согледувања	83
9.1	Споредба со други резултати	83
9.2	Осврт на постигнати резултати	84
9.3	Заклучок	85
9.4	Придобивки од добиените модели	86
9.5	Идни истражувања	87

Листа на слики

2.1	Концепти кај облачето	18
2.2	МОСНА архитектура	20
2.3	Архитектура на пресметување во магла	21
2.4	Архитектура на Dew пресметување	22
4.1	Tesseract OCR процесирање	31
4.2	Детектирање на косина на сликата	33
4.3	Бинаризација на сликата	33
4.4	Сегментација на карактерите во линија	34
4.5	Поединечна сегментација на карактерите	34
4.6	Извлекување на карактерите од сликата	35
4.7	Повеќе слојна перцепција	35
4.8	Комуникациски и процесирачки активности за OCR процесирање на слика сликана со мобилен телефон	36
4.9	Сценарио L: Процесирање на слика и OCR процесирање во облаче	37
4.10	Сценарио F: Процесирање на слика и OCR процесирање во магла	37
4.11	Сценарио C: Процесирање на слика и OCR процесирање во облак	38
4.12	OCR Апликација	39
5.1	Примери на различност а) Различност 1, б) Различност 2	44
5.2	Комплексност на сликата а) нормална, б) закосена, в) заматена	45
6.1	Времињата на одговор кај различните сценарија користејќи мали слики	49
6.2	Времињата на одговор кај различните сценарија користејќи големи слики	49
6.3	Време за OCR процесирање - Мобилен уред	50
6.4	Време за OCR процесирање - Облаче	50
6.5	Време за OCR процесирање - Магла	51
6.6	Време за OCR процесирање - Облак	51
6.7	Забрзување помеѓу Закосена и Нормална комплексност	52
6.8	Забрзување помеѓу Заматена и Нормална комплексност	52
6.9	Забрзување $S_{variety}$ кај нормална комплексност на сликата	53
6.10	Забрзување $S_{variety}$ кај закосена комплексност на сликата	53
6.11	Забрзување $S_{variety}$ кај заматена комплексност на сликата	54
6.12	Распределба на точноста во однос на бројот на карактери на сликата	54

6.13	Забрузвање спроти мобилното решение за мали слики	56
6.14	Забрузвање спроти мобилното решение за големи слики	56
6.15	Забрузвање помеѓу различните сценарија за мали слики	57
6.16	Забрузвање помеѓу различните сценарија за големи слики	57
7.1	Определување на наклонот (α) vs N_C	63
7.2	Дериват на времето на одговор (наклон) β vs N_C и IS	64
7.3	Тренд на измерените вредности и моделираната функција	65
7.4	Приближување на моделираните функции на зависност на времето на одговор од големината на сликата	65
8.1	Модел на време за одговор, вклучувајќи време на комуникација и процесирање за сценарио L - облаче	77
8.2	Модел на време за одговор, вклучувајќи време на комуникација и процесирање за сценарио магла (3G)	77
8.3	Модел на време за одговор, вклучувајќи време на комуникација и процесирање за сценарио магла (4G)	77
8.4	Модел на време за одговор, вклучувајќи време на комуникација и процесирање за сценарио C - облак	78
8.5	Време за процесирање на различните сценарија за мали слики	78
8.6	Време за процесирање на различните сценарија за големи слики	79
8.7	Релативна грешка помеѓу различните сценарија за мали слики	79
8.8	Релативна грешка помеѓу различните сценарија за големи слики	80
9.1	Canvas бизнис модел	88

Листа на табели

2.1	Разлики помеѓу облак и облаче	19
2.2	Разлики помеѓу магла и роса	23
5.1	Техничка спецификација на тестната околина	42
5.2	Број и големина на карактери користени во тестните сценарија	43
6.1	Време на процесирање на сликата и препознавање на карактери користејќи различна архитектура	48
6.2	Однос $Acc_{variety1}/Acc_{variety2}$ за различна комплексност на слики	55
6.3	Однос помеѓу точноста кај закосени и заматени спроти нормални слики .	55
6.4	Анализа на потрошувачката на батерија	58
8.1	Времиња на одговор за секоја дефинирана активност	71
8.2	Типичен проток на податоци V и задоцнување D за Wi-Fi/3G/4G/WAN мрежи	73
8.3	Најдобри и најлоши случаи за мрежно задоцнување	73
8.4	Процесирачка моќ на мобилните уреди користени во експериментите . .	74
8.5	Процесирачка моќ на серверите од средна класа користени во експери- ментите	74
8.6	Најдобро и најлошо сценарио за процесирање на слика и препознавање на карактери (во секунди)	76
8.7	Најдобри и најлоши пресметувачки резултати (во секунди)	76

Глава 1

Вовед

Бројката на корисници коишто имаат пристап до Интернет денес е фасцинантна и достигнува и до 5 милијарди. Мобилните уреди се важен дел од секојдневието на луѓето и претставуваат најважна и најсоодветна комуникациска алатка. Корисниците може да користат голем број на мобилни апликации (iPhone apps, Google apps, Windows Phone apps, etc.) на своите мобилни уреди. Со подобрување на карактеристиките на мобилните телефони (процесорот, меморијата, поврзаноста) луѓето се повеќе ги користат истите за праќање на мејлови, Интернет банкарство, навигација, играње, сликање, снимање, итн. Отвореноста на оперативните системи доведе до развој на голем број на мобилни апликации и овозможи поголема интеграција на истите. Постоечките мобилни апликации, како читачи на бизнис картички, апликации за потпишување, апликации за преведување, им помагаат на луѓето брзо да ја добијат информацијата во дигитален формат. Некои од тие апликации вклучуваат OCR (Optical Character Recognition) алгоритам, кој претставува моќна алатка за претворување на информацијата од аналогна во дигитална. Користењето на оваа технологија може да биде од корист во креирањето на дигитални библиотеки, препознавање на текст од слики, препознавање на рачно напишани документи, преведување, итн. Има голем број на OCR базирани апликации за десктоп компјутери, но само неколку кои се применливи и за мобилните уреди.

Бројот на апликации наменети за потребите на корисниците на мобилни уреди, секојдневно се зголемува. Заради ограниченоста на сопствените ресурси, мобилните уреди имаат потреба од користење на дополнителни ресурси. Алтернативна метода е користење на надворешни процесирачки единици, кои им дозволуваат на крајните корисници да ги користат нивните ресурси за извршување на мобилните апликации.

Една од најпопуларните методи е користењето на ресурсите на облакот. Облакот игра улога на сервер кој им овозможува на корисниците да ги користат неговите ресурси за извршување на апликации кои бараат ресурси од мобилниот уред којшто во тоа време не ги поседува. Ресурсите на облакот им овозможуваат на корисниците да ја зголемат процесирачката моќ на мобилниот уред, во зависност од потребите во даден момент [1], [2]. Сепак, користењето на ресурсите на облакот за процесирање на мобилни апликации

може да биде лоша одлука. Облакот се наоѓа далеку од мобилниот уред и WAN (Wide Area Network) доцнењата го прават овој пристап неефикасен за апликациите кои се извршуваат во реално време.

Решение за ваквиот проблем е пристапот со користење облаче воведен од Satya-narayana [3], кој има голема предност при пресметувањето во облакот без WAN ограничувања. Облачето претставува мал сервер кој што се наоѓа во близина на мобилниот уред. Тој е сигурен и моќен компјутер способен да ја извршува работата како и секој оддалечен облак. Поради близината на облачињата, овој пристап овозможува побрзо извршување на апликациите на многу поедноставен начин. Овој концепт им овозможува на мобилните корисници да ги користат компјутерите во близина, за да ги искористат предностите на процесирањето во облакот но без WAN доцнења, загушувања или испади.

Покрај овие два моментално најкористени пристапи, во последно време се поактуелни се пристапите со пресметување во магла и пресметување во роса. Пресметувањето во магла ги пренесува податоците поблиску до крајните корисници и со тоа го елиминира задоцнувањето при пренесување на податоците. Предноста кај овој пристап е тоа што серверите кои го прават процесирањето се позиционирани во самата базна станица на мобилниот оператор.

Пресметувањето во роса го надминува концептот на мрежа/простор/сервис и го дистрибуира процесирањето на таканаречена под-платформа. Тоа се базира на концептот на микро сервиси во вертикално дистрибуирана пресметувачка хиерархија. Споредбено со пресметувањето во магла кое ги поддржува новите IoT (Internet of Things) апликации кои се извршуваат во реално време, со предвидено задоцнување и динамичко мрежно преконфигурирање, пресметувањето во роса ги поместува границите на апликациите, податоците и сервисите на основно ниво надвор од централните виртуелни јазли кон крајните корисници [4].

Предностите на овие процесирачки единици ги направија истите можни, а во некои случаи и задолжителни процесирачки единици за OCR извршување на мобилните платформи.

1.1 Мотивација

Брзиот раст на пазарот на мобилни телефони, ја наметна потребата од голем број на апликации кои се развиени за извршување на мобилните уреди. Мобилните уреди немаат доволно ресурси за извршување на голем дел од ваквите апликации, или пак нивното извршување е многу бавно и не е прифатливо за самите корисници. Во одредени случаи, уредите имаат ограничени ресурси што резултира со попреченост во извршувањето на самиот мобилен уред. Поради тоа потребни се дополнителни ресурси, за да извршувањето на ваквите апликации биде побрзо [5].

Постојат голем број апликации кои користат дополнителни ресурси за процесирање и презентирање на резултатите назад на мобилниот уред. Посебно се важни аплика-

циите кои се извршуваат во реално време. Едни од тие, се апликациите кои користат OCR алгоритам за процесирање и препознавање на текст. OCR алгоритмот може да се дефинира како електронска анализа на сликата за да ги идентификува деловите кои содржат текст и извлече и препознае карактери од истата. Со помош на скенер се креира дигитална слика од хартиени документи. Ако скенираните слики се со добра резолуција, тогаш процесирањето и извлекувањето на текстот е полесно. Резолуцијата од 300dpi или повеќе е обично доволна за оптичко препознавање на карактерите (OCR).

Со зголемувањето на достапноста на дигиталните камери и мобилните уреди, се зголеми и интересот за користење на OCR базирани апликации. Потребата од вакви апликации кои ќе им овозможат на луѓето да сликаат, препознаваат и преведуваат различни јазици сè повеќе се зголемува. Тие можат да им овозможат на корисниците да го разбираат било кој јазик, како и да добијат информации за земјата во која се наоѓаат поврзани со музеите, рестораните, хотелите, културните збиднувања и останати корисни информации.

OCR апликациите веќе се користат кај десктоп компјутерите. Сликите мора да се постават мануелно или да бидат сликани директно со веб камерата на десктоп компјутерот. Резолуцијата на сликите сликани со веб камера е само 1.3 мегапиксели, па квалитетот на OCR препознавањето може да биде лош. Од друга страна пак, точноста на OCR препознавањето на слики од мобилен уред е многу подобра поради резолуцијата од 5 или повеќе мегапиксели што ја поседуваат камерите на мобилните уреди. Сепак, процесирањето на сликите на мобилни уреди е ново поле за истражување со многу предизвици поради ограничениот хардверски лимит, ограничениот процесор, квалитетот на сликите и поврзаноста. Мобилните уреди со висококвалитетни камери, моќни процесори и голема меморија се сè поприсутни. Најголем дел од постоечките OCR апликации за мобилни уреди се извршуваат на оддалечени сервери кои немаат проблеми поврзани со ограничена меморија или процесирачка моќ, но сепак имаат проблеми поврзани со задоцнување во комуникацијата. Поради тоа, предизвик е да се најде решение кое ќе ги користи и мобилниот уред и серверот како процесирачка единица за извршување на OCR алгоритмот и добивање на оптимални резултати. Бидејќи локалното процесирање може да одземе многу време, потребно е решение кое ќе биде многу побрзо и кое ќе ја користи локалната интернет конекција. Тоа беше предизвик да се најде независно решение кое нема да биде инсталирано на мобилниот уред (бидејќи би барало јаки перформанси на истиот како меморија или батерија) и да не биде инсталирано на далечниот облак (за да се избегне долгото WAN задоцнување при трансфер на сликата).

1.2 Предмет на истражувањето

Тековните методологии за развој на мобилни апликации кои користат надворешни ресурси за извршување (облак, облаче, магла, роса) сеуште не се усовершени до таа мера да целосно се користат во развојот на денешните мобилни апликации. Постојат голем

број недостатоци и нејаснотии кои се главна причина што имплементацијата на ваквите пристапи е сеуште во рана фаза.

Предмет на истражување на оваа докторска дисертација е анализа и имплементација на пристапите за пресметување во облаче, пресметување во магла и пресметување во роса за апликација на мобилните телефони, која опфаќа алгоритам за оптичко препознавање знаци, зборови и фрази и наоѓање на оптимално решение.

Точноста, флексибилноста и брзината на извршување на процесирањето, како и однесувањето на OCR алгоритмот се основните параметри коишто се анализираат во четирите одвоени случаи. Бидејќи алгоритмите за препознавање знаци можат да бидат многу сложени, а со тоа и доста бавни при извршувањето, истражени се начини за нивно забрзување користејќи соодветни пристапи на комбинирање на алгоритмите, но и со помош на дистрибуирано и паралелно пресметување. Притоа се употребуваат општи платформи кои не користат специјализиран хардвер и не се специјализирани за конкретен домен, а може лесно да се надоградуваат заедно со растот на податоците.

Истражувањето вклучува неколку аспекти при извршување на процесирањето на знаците. Најпрво се анализираат постоечките методи кои се користат кај алгоритмите за оптичко препознавање знаци. Потоа со помош на тие методи се имплементира алгоритмот да може да се извршува во облак, облаче, магла или роса. На крај се прави анализа од споредбата на сите пристапи, се дефинираат предностите и недостатоците на секој од нив и се даваат препораки за подобрување на алгоритмите за препознавање знаци како и техниките кои се користат при препознавање кај истите.

1.3 Цели на истражувањето

Главна цел на докторската дисертација е да моделира однесување на мобилна апликација којашто користи алгоритам за оптичко препознавање знаци, зборови и фрази, со користење на различни архитектурни решенија за пресметување во облак, облаче, магла или роса и да даде одговор кое решение е оптимално.

Алгоритмот за оптичко препознавање знаци се извршува на оддалечен сервер, независно дали е облак, облаче, магла или роса. Во насока на таа цел, преку низа истражувања и анализи е одбран алгоритмот кој ги исполнува условите за имплементација на трите пристапи. Дефинираниот модел има можност да препознава доста сложени знаци и симболи внесени преку мобилна апликација. Истражувањето е извршено поединечно и истовремено на оддалечените облак, облаче, магла и роса.

За да се постигне главната цел, дефинирана е следнава главна хипотеза:

H0: Употребата на архитектура за пресметување во облаче и пресметување во магла може во голема мера да го подобри извршувањето на мобилни апликации споредбено со користење на архитектура за пресметување во облак или извршување на апликацијата на мобилниот телефон.

Дополнително се дефинирани и следниве хипотези:

- H1: Примената на пресметувањето во облаче и пресметувањето во магла дозволува побрз трансфер на податоци во однос на другите разгледувани архитектури и решенија
- H2: Примената на пресметувањето во облаче и пресметувањето во магла резултира со побрзо процесирање на апликацијата за препознавање знаци во однос на другите разгледувани архитектури и решенија
- H3: Примената на пресметувањето во облаче и пресметувањето во магла резултира со помала искористеност на ресурсите на мобилниот уред на апликацијата за препознавање знаци во однос на другите разгледувани архитектури и решенија

Покрај зададени хипотези, дефинирани се и следните истражувачки прашања:

- R1: Како зависат перформансите од големината на сликата?
- R2: Како зависат перформансите од бројот на карактери?
- R3: Како зависат перформансите од комплексноста на сликата?
- R4: Колкава е точноста на препознавање на карактери?
- R5: Како делува архитектурното решение на перформансите?
- R6: Кое од претставените архитектурни решенија е најдобро?
- R7: Како зависи потрошувачката на батерија при обработка на сликата и препознавање на карактери?
- R8: Како се моделираат перформансите на алгоритмот во зависност од карактеристиките на сликата?
- R9: Како се моделираат перформансите на системот во зависност од различните архитектурни решенија?

Крајната цел е да се споредат добиените резултати и да се донесе заклучок кој од сите анализиани пристапи е најдобар, односно да се одредат домените каде соодветниот пристап ќе има најефикасна примена. За реална применливост на заклучоците од истражувањето потребен е соодветен бизнис модел. Една од целите на оваа докторска дисертација е да се дефинираат и можни бизнис модели кои би се имплементирале и користеле од корисниците на мобилни услуги.

1.4 Придонес од истражувањето

Главен придонес од оваа докторска дисертација е во тоа што ги одредува домените каде соодветниот разгледуван пристап за реализација на мобилна апликација за препознавање на знаци има најефикасна примена. Пристапите користени во оваа дисертација ќе ги прошират тековните истражувања во оваа област во насока на примена на разгледуваните технологии за надворешно процесирање. Во процесот на анализирање на овој модел се анализираат предностите и недостатоците од користењето на надворешни ресурси за извршување на мобилни апликации кои побаруваат големи ресурси, како на пример интензивни пресметки, голема меморија, батерија итн.

Предноста на разгледуваните пристапи за надворешно пресметување во облак, облаче, магла и роса е да се заштедат ресурсите на мобилниот телефон и да овозможи зголемување на бројот на мобилни апликации во комуникацијата на корисниците на мобилни уреди. Постојат голем број на мобилни апликации кои побаруваат големи ресурси, но многу мал дел од нив се направени нивното извршување да биде надвор од мобилниот уред. Извршувањето на ваквите апликации на самиот мобилен уред е многу голем предизвик со оглед на процесирачката моќ на мобилниот уред, меморијата, потрошувачката на батерија, итн. Поради тоа со помош на истражувањето направено во оваа докторска дисертација, дефиниран е оптимален пристап којшто овозможува извршување на апликација што користи оптичко препознавање на знаци со примена на надворешен ресурси (облак, облаче, магла, роса).

Апликациите кои бараат надворешно процесирање се од голема корист како за самите корисници, така и за поголемите компании. Со користењето на таквите апликации може да се врши фотографирање на одреден знак, симбол, писмо и да се прикаже нивното значење назад на мобилниот уред. Исто така, ваквиот пристап може да најде примена при фотографирање на одредени ликови, знаци и нивно препознавање со што крајниот корисник би добил информација за истите.

Мобилните апликации од ваков тип долго време се користат со веб базиран пристап. Но со денешниот напредок на технологијата, неговата примена сè повеќе се префрлува на мобилните уреди преку специјално дизајнирани апликации. Со пристапот предложен во оваа дисертација би се олеснил начинот на којшто луѓето би ги користеле апликациите за препознавање знаци бидејќи одговорот на бараната информација би го добиле во реално време на мобилниот уред, без големо задоцнување и со многу добар квалитет, а со самото тоа би се зголемило задоволството кај крајниот корисник.

Глава 2

ОСНОВНИ КОНЦЕПТИ

Истражувањето направено во оваа докторска дисертација ќе им помогне на корисниците на мобилни уреди да сликаат и препознаваат текст од јазик кој што не им е познат, преку користење на оддалечени сервери како процесирачки единици, а со тоа штедење на ресурсите на својот мобилен уред. Во продолжение се опишани најновите технологии коишто се користат за таа цел, како што се OCR алгоритмот, пресметување во облак, облаче, магла и роса.

2.1 OCR (Оптичко препознавање на карактери)

OCR е технологија која овозможува претворање на скенирани документи, PDF документи и слики сликани со дигитална камера во текст којшто може да се обработува. Процесирањето кај OCR започнува со анализа на скенираната слика за да се идентификуваат буквите и бројките, а потоа следи препознавање на карактерите и нивно претворање во ASCII код.

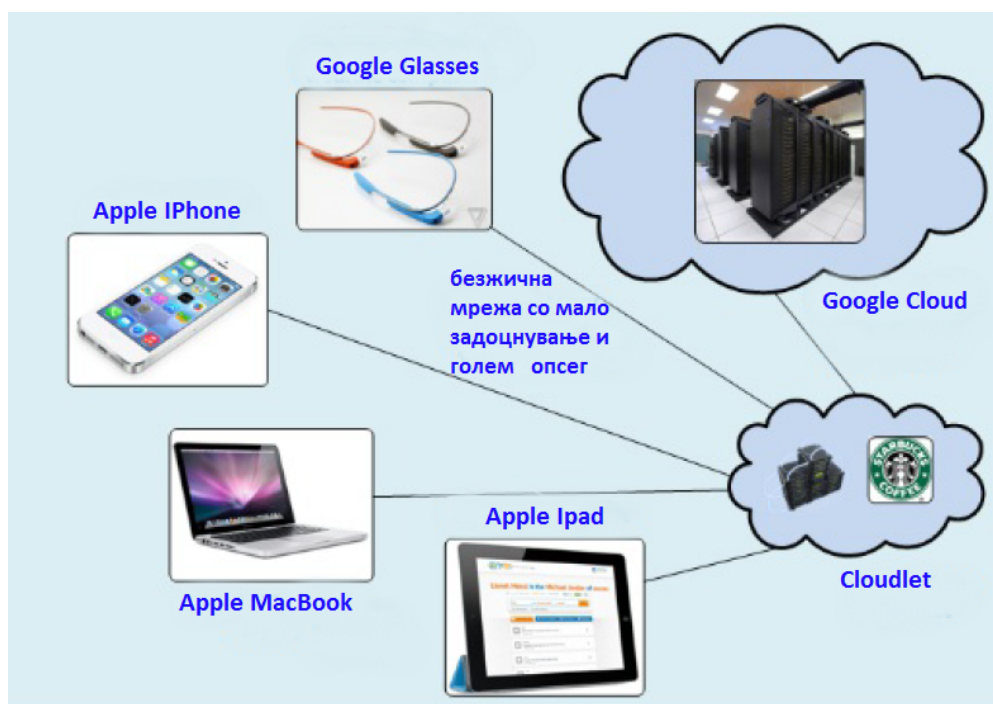
OCR алгоритмот е широко распространет и се користи за скенирање на официјални документи за дигитално архивирање, списанија, бизнис картички, пасоши, знаци, итн. Постојат многу методи и OCR решенија но само неколку се бесплатни. Скоро сите OCR системи се наменети за специјални типови на слики и комбинираат различни оптимизирачки методи за подобрување на точноста на препознавање.

2.2 Пресметување во облаче

Бројот на корисници кои користат мобилен уред за да се поврзат на Интернет константно се зголемува. Познато е дека мобилните уреди имаат послаби перформанси споредбено со статичките сервери, како времетраењето на батеријата, меморијата и процесирачката моќ на самиот уред [6].

Едно од решенијата за намалување на проблемите со ограничените ресурси на мобилните уреди е пресметувањето во облак. Но сепак, големите WAN задоцнувања се клучен недостаток во интеракцијата на мобилниот уред со серверот. Поради тоа пот-

ребна е имплементација на облаче, помал сервер кој ќе биде лоциран во близина на мобилниот корисник кој ќе се поврзе преку LAN мрежа со оддалечените сервери на облакот [7]. Облачињата претставуваат децентрализирани и широко распространети Интернет инфраструктурни компоненти чии ресурси може да се користат од мобилните уреди кои се наоѓаат во близина како што е претставено на Слика 2.1 [8]. Во суштина тие претставуваат мали податочни центри кои се само-менаџираат, трошат малку електрична енергија, Интернет поврзување и корисничка контрола. Облачето наликува на кластер од повеќејадрени компјутери со гигабитен Интернет и широкоокопојасен LAN (Local Area Network). Табела 2.1 подетално ги претставува разликите помеѓу облакот и облачето.



Слика 2.1: Концепти кај облачето

2.3 МОСНА архитектура

МОСНА (MOBILE Cloud Hybrid Architecture) архитектурата е креирана како решение за поддршка на големи мобилни апликации кои го користат облакот за процесирање [9], [10], [11], [12]. Таа се состои од три дела: мобилен уред, облаче и облак. Во докторската дисертација мобилниот уред се користи за сликање на слики и испраќање на истите на облакот или облачето во формат за обработка. Процесирањето и препознавањето на карактерите на сликата се прави на облакот или облачето и резултатот се враќа назад на мобилниот уред. МОСНА архитектурата е претставена на Слика 2.2

Ова претставува еден од многуте предизвици за употреба на надворешна инфраструктура при користење на апликации на самиот мобилен уред. Имајќи го во предвид

Табела 2.1: Разлики помеѓу облак и облаче

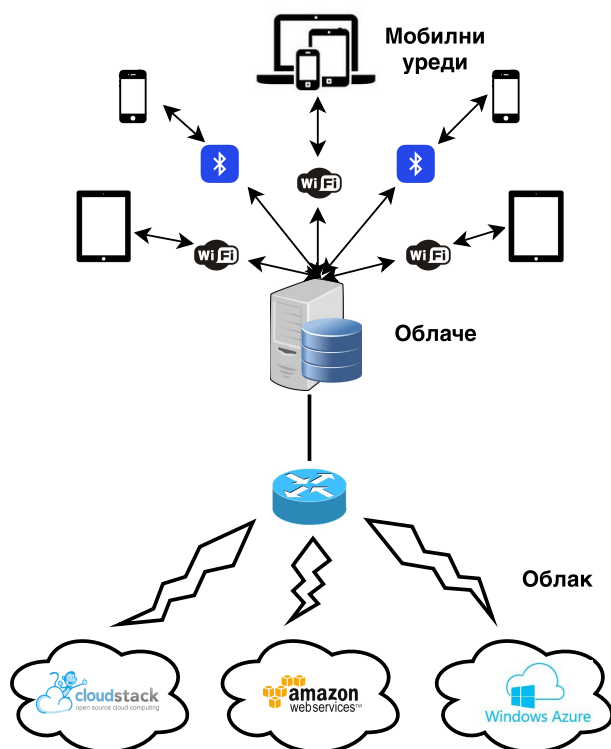
	Облаче	Облак
<i>Состојба</i>	Привремена состојба (термин кој се користи за информации кои истекуваат (исчезнуваат) доколку не се освежени односно кеширани копии од податоци или код кои се достапни и на други места)	Постојана (состојба во која постои само една единствена копија од податоци или код) и привремена состојба
<i>Администрирање</i>	Само-управуван - професионално внимание не е потребно или е малку потребно	Администриран од професионалци, оператори кои работат 24x7
<i>Локација</i>	”Центар за податоци во кутија” кој може да е сместен во канцелариите	Машинска соба со соодветен систем за ладење и греење
<i>Големина</i>	Сервер	Автоматизиран серверски систем со контрола на енергија и температура
<i>Совместивост</i>	Децентрализирана од локалниот бизнис	Централизирана - кај големи провајдери
<i>Поврзаност</i>	LAN поврзување	Wi-Fi конекција
<i>Капацитет</i>	Неколку корисници истовремено	100-1000 корисници истовремено

овој предизвик, употребен е Tesseract алгоритмот користејќи ја МОНА архитектурата за процесирање. Горниот дел од Слика 2.2 ја претставува архитектурата на облачето каде мобилниот уред директно е поврзан и разменува податоци со локалниот сервер на облачето користејќи Wi-Fi поврзување, без да го контактира серверот на облакот. Облачето и облакот се поврзани меѓу себе само во случај кога се прави синхронизација и размена на податоци.

Главна прашање е како да се дистрибуира процесирачкото оптоварување, а со тоа да се добие минимално време на одговор кога мобилниот уред комуницира со повеќе сервери на облакот. За таа цел се користи Андроид базиран паметен телефон како главен уред со којшто се сликаат сликите и се испраќаат директно на облачето за процесирање. Облачето го прави препознавањето и го враќа резултатот назад на мобилниот уред.

2.4 Пресметување во магла (Fog computing)

Мобилните апликации за секојдневна употреба користат ресурси кои го зголемуваат мобилниот податочен сообраќај. Пренесувањето на огромни количини на податоци на



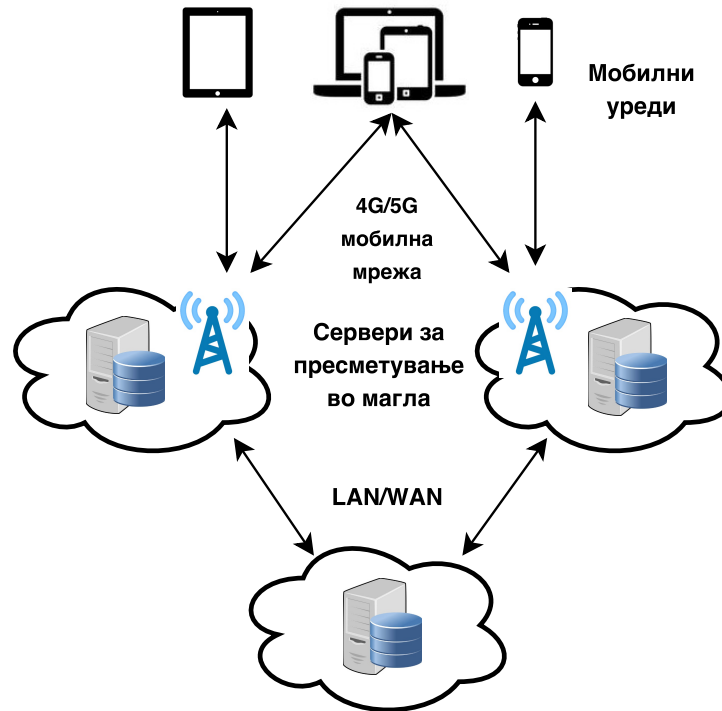
Слика 2.2: МОСНА архитектура

облакот и назад на мобилниот уред бара висок пропусен опсег за да се избегне задоцнувањето. Мобилните оператори пронајдоа нов начин на обезбедување на дополнителни услуги на додадена вредност на нивните корисници со овозможување на ресурси на крајот на нивната мрежа, и тоа не само за комуникациски услуги туку и за основно процесирање на податоци [13].

Пристапот на пресметување во магла е од голема предност поради позиционирањето на серверите во близина на базната станица на мобилниот оператор, со тоа справувајќи се со зголемениот податочен сообраќај, апликациите, хостираните сервиси на крајот на мрежата [14]. Оваа архитектура ги пренесува податоците поблиску до крајните корисници со тоа елиминирајќи го задоцнувањето, како и бројните чекори и го поддржува пренесувањето на податоци и мобилното пресметување [15]. Пристапот на пресметувањето во магла го забрзува извршувањето на апликациите за препознавање карактери и слики во реално време и со тоа го избегнува задоцнувањето во мрежата и овозможува побрз одговор на мобилниот уред пресметан во милисекунди. Со ова се заштедува време, процесирачка моќ и секако батерија на мобилниот уред. Ова е особено важно за апликации кои се извршуваат во реално време како што се OCR апликациите кои побаруваат голема процесирачка моќ и брз одговор на барањата назад на мобилниот уред.

Слика 2.3 ја прикажува архитектурата на пресметување во магла, што може да биде еден од можните решенија на проблемот на истражување. Разликата од претходната МОСНА архитектура која користи процесирање во облаче е користењето на Интернет

поврзување од мобилниот оператор наместо користење на Wi-Fi или Bluetooth поврзување. За разлика од решението со користење на облаче за процесирање, мобилниот оператор ги користи серверите на крајот на мрежата. Крајните сервери се позиционирани во базната станица што комуницира со мобилниот уред, а по потреба го користи и главниот сервер што се наоѓа во облакот на мобилниот оператор. Овој пристап ги доближува ресурсите до крајниот корисник на крајот на мрежата на мобилниот оператор.



Слика 2.3: Архитектура на пресметување во магла

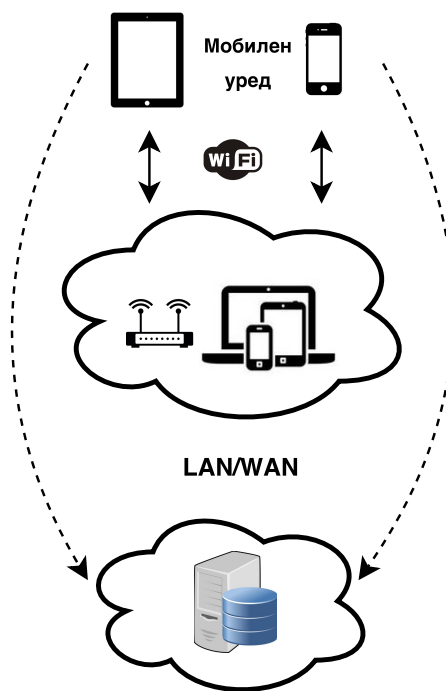
2.5 Пресметување во роса (Dew computing)

Пресметувањето во роса е нова парадигма која се појави после широката прифатеност и употреба на пресметувањето во облак. Идејата за пресметувањето во роса е да се користат ресурсите колку што е можно повеќе пред процесирањето да се префрли на серверите на облакот. Архитектурата на пресметување во роса овозможува користење на микро сервиси во соработка со макро сервиси или пак сервиси во роса во соработка со сервиси на облакот [13].

Независноста кај пресметувањето во роса значи дека е можно користење на дадени сервиси без да се користат сервисите на облакот и без Интернет поврзување. Исто така, постои соработка помеѓу апликациите кои користат пресметување во роса и кои мора автоматски да разменуваат информации со сервисите на облакот за време на нивното извршување. Ова претставува одлично решение за апликации во реално време кои користат сервиси од облакот, бидејќи пресметувањето во роса може да се користи во случаи кога мобилниот уред нема Интернет поврзување.

Услугите на пресметувањето во роса може ефикасно да се користат за IoT апликации каде голем број на сензори овозможуваат проток на податоци до локалниот концентратор. Овие мали сервери за пресметување во роса обезбедуваат локализирани микро сервиси, како собирање и чување на влезниот проток на податоци и локално процесирање. Исто така тие ги испраќаат резултатите на облакот или алармираат ако има потреба од брза реакција. Класичен пример претставува, пренесување на видео снимки од камери и микро сервиси кои овозможуваат контрола на сообраќајот.

МОСНА архитектурата всушност ги унифицира концептите на пресметување во облаче и пресметување во магла со воведување на помали сервери споредбено со податочните центри во облакот и овозможува побрз пристап и дистрибуирање на процесирањето на овие сервери. Разликата на пристапот со пресметување во магла и МОСНА архитектурата е користењето на локални концентратори коишто овозможуваат користење на микро-сервери за локално процесирање. Тие може да работат независно дури и без Интернет поврзување. Концентраторите на пример, може да се специјализирани уреди во "паметна" домашна средина или решенија за контрола на сообраќајот со видео камери кои го скенираат сообраќајот. На Слика 2.4 е претставено такво решение каде локалниот концентратор претставува мала процесирачка единица. Во "паметните" домови, тоа може да е додаток на Wi-Fi рутерот или на мобилниот уред којшто ги поврзува различните сензори кои овозможуваат OCR микро сервиси и се сместени во близина на WiFi рутерот. Табела 2.2 подетално ги претставува разликите помеѓу пресметувањето во магла и пресметувањето во роса.



Слика 2.4: Архитектура на Dew пресметување

Табела 2.2: Разлики помеѓу магла и роса

	Магла	Роса
<i>Состојба</i>	Привремена состојба (термин кој се користи за информации кои истекуваат (исчезнуваат) доколку не се освежени односно кеширани копии од податоци или код кои се достапни и на други места)	Постојана (состојба во која постои само една единствена копија од податоци или код) и привремена состојба
<i>Администрирање</i>	Само менаџирање со мала и непрофесионална администрација	Непрофесионална администрација
<i>Локација</i>	Во базната станица на мобилниот оператор	Додаток на Wi-Fi рутерот или мобилниот телефон
<i>Големина</i>	мал сервер	микро-макро сервиси сместени на мали сервери
<i>Сопственост</i>	На мобилниот оператор	На самиот сопственик на мобилниот уред
<i>Поврзаност</i>	3G/4G поврзување	Wi-Fi поврзување
<i>Корисници</i>	1-100 корисници истовремено	1-10 корисници истовремено

2.6 Бизнис модел на решението

Почетната точка за секоја дискусија, состанок или работилница за иновацијата на бизнис модел, е запознавање со тоа што навистина е бизнис модел. Бизнис моделот треба да биде едноставен, релевантен и разбирлив за сите. Canvas претставува рамка за стратешки менаџмент за развој на нови или постоечки бизнис модели составена од 9 блока претставени подолу:

- *Customer Segments - Кориснички сегменти* - Корисничкиот сегмент ја одредува целната група која што би ја користела апликацијата за препознавање на букви, знаци, бројки, која вклучува посетители на објекти (ресторани, трговски центри, аеродроми), да одберат кој пристап ќе го користат за извршување на апликацијата за да го дознаат значењето на зборови, изрази, на јазик кој не го познаваат.
- *Value Proposition - Предложена вредност* - Вредноста на продуктот е дефинирана преку неколку пристапи кои го издвојуваат решението од сите преостанати на пазарот со тоа што извршувањето на апликацијата дава иновативен пристап и овозможува извршување на облак, облаче, магла и роса. Тоа овозможува едноставно користење на производот од страна на корисникот не земајќи ги во предвид различната имплементација на истиот во позадина.
- *Channels - Канали на дистрибуција* - Каналот за дистрибуција на OCR базираната апликација е преку Play Store на Google. Исто така постои можност да се импле-

ментира канал на дистрибуција да биде рор-уп страна при логирање на бесплатна Wi-Fi мрежа во угостителски објекти, трговски центри, аеродроми, каде корисниците ќе можат бесплатно да ја преземат апликацијата.

- *Customer relationships - Однос со корисници* - Апликацијата нуди пристапен бесплатен автоматизиран сервис за самостојно користење за лична поддршка за корисници кои сакаат да одберат кој пристап ќе го користат за извршување на апликацијата.
- *Revenue Streams - Прилив на средства* - Користењето на апликацијата е бесплатно за основните функционалности при извршување на апликацијата. Дополнителен извор на генерирање на приходи може да биде рекламниот простор отстапен за користење на клучните партнери.
- *Key Resources - Клучни ресурси* - Клучни ресурси се човечките кои работат на одржување и администрирање на апликацијата.
- *Key Activities - Клучни активности* - Клучните активности кои се потребни за да се добие предложената вредност на производот се дефинирани во овој блок. Исто така се дефинира кои клучни активности ќе бидат преземени користејќи ги дистрибуциските канали, каков ќе биде односот со корисниците и колкав ќе биде приливот на средства од овие активности.
- *Key Partnerships - Клучни партнерства* - Клучните партнерства се дефинирани преку соработка со субјектите кои поседуваат инфраструктура за поддршка на решението и негова имплементација. Тие преземаат активности со кои овозможуваат рекламирање на апликацијата, како извор на дополнителен прилив на средства.
- *Cost Structure - Структура на трошоци* - Во структурата на трошоците дефинирана во бизнис моделот најголем дел зафаќа трошокот кој вклучува ресурси за развој на производот, негово одржување (доколку има потреба) и развој на дополнителни функционалности по барање на корисникот.

2.7 Споредба на архитектурните пристапи на ОСР алгоритмот со мобилно пресметување

Сите архитектурни пристапи обработени во докторската дисертација имаат различен начин и пристап кон извршувањето на барањата. За да се зачуваат ресурсите на мобилниот уред и да се зголеми траењето на батеријата, извршувањето се пренесува на оддалечен сервер во зависност од локацијата и можноста. Доколку пренесувањето се прави на облаче тогаш се користи Wi-Fi поврзување, додека, доколку извршувањето се прави во магла тогаш се користи 3G/4G поврзување и брзината на истото во голема мера зависи од брзината на 3G/4G поврзувањето. Идејата пак кај пресметувањето во роса е да се користат ресурсите колку што е можно повеќе пред процесирањето да се префрли на облакот. Користењето на овие архитектурни пристапи овозможува целосно префрлување на пресметувањето и со тоа му овозможува на мобилниот уред зачувување на сопствените перформанси.

Глава 3

Преглед на тековни истражувања

Секојдневно се прават нови истражувања во насока на подобрување на алгоритмите за OCR препознавање и воведување на нови архитектури кои би го забрзале истото. Поради тоа направена е анализа и споредба на тие истражувања со истражувањата направени во оваа дисертација и истите се претставени подолу.

3.1 Примена на OCR кај мобилни уреди

Со масовната употреба на дигиталните камери, адаптацијата на алгоритмите за препознавање на текст стана многу популарна. Направени се големи напори овие алгоритми да се прилагодат за користење од страна на мобилните уреди. Истражувачите веќе имаат предложено некои примери на самостојни OCR системи кои се извршуваат на мобилните уреди [16], [17]. Senda и останатите презентираат интерфејс кој се потпира на концентрацијата на повеќе слики направени од камерата на мобилниот уред [18]. Системот изграден од Laine и Nevalainen [19] најголем дел од времето за пресметување посветува на извршување задачи кои се специфични за адаптација на текстот за препознавање за мобилни околина (на пример, приспособување на закосувањето на текстот).

Постојат голем број на OCR решенија достапни бесплатно за користење. Некои од нив се, GOCR [20], OCRAD [21], OCROpus [22], OpenOCR [23], Tesseract [24], OCRDroid [25], итн. Исто така постојат и некои комерцијално достапни OCR системи како ABBYY Fine Reader [26], OmniPage [27], Microsoft Office Document Imaging, ExperVision Open-RTK [28], итн.

3.2 Архитектурни пристапи

Постојат неколку архитектурни пристапи кои се користат при процесирање со OCR алгоритрам. Повеќето од решенијата се базирани на процесирање и препознавање карактери од слики. Главниот предизвик е каде се прави процесирањето. Кога процесирањето се прави на самиот мобилен уред нема некои големи разлики во времињата за враќање на резултатот назад на мобилниот уред. Но кога сликата треба да се префрли

на најблиското облаче или облак тогаш доцнењата во мрежата и протокот на податоци ја прават главната разлика. Garcia и останатите објаснуваат во [29] како различното време на задоцнување значително влијае на резултатите. Исто така, Caushaj и останатите објаснуваат во [30] дека задоцнувањето и протокот на податоци зависат од архитектурата, колку што помал број на компоненти се вклучени во архитектурата толку се подобри целосните перформанси на мрежата.

Задоцнувањето и потрошувачката на енергија при пресметувањето во облаче, магла или роса се главните параметри кои голем број истражувачи се обидуваат да ги подобрат и доведат до совршенство.

3.2.1 Пресметување во облаче

Облачето може да обезбеди слободни ресурси на блиските мобилните уреди со побрз пристап и помала потрошувачка на енергија. За да се стимулира користење на услугите на облачето потребен е исплатлив и ефикасен механизам. Jin и останатите во [31] предлагаат вистински поттикнувачки механизам (truthful incentive mechanism (TIM)) за координирање на ресурсите помеѓу мобилните уреди како сервисни уреди и облачињата како сервисни провајдери. Овој механизам овозможува таканаречена двојна аукција во која и сервисните уреди и сервисните провајдери ги поднесуваат своите понуди и побарувања соодветно. Индивидуалната рационалност гарантира дека на сервисниот уред никогаш нема да му биде наплатено повеќе од што понудил, додека на сервисниот провајдер нема да му биде платено помалку од што побарал.

Во друго истражување облачето се користи како технологија за учење. Во [32] е претставен прототип на пристап за користење на облачето за мобилно отворено и далечинско учење (Open and Distant Learning (ODL)). Мобилниот слушател иницира барање за поврзување кое агентот го процесира за да го одреди минималното задоцнување и протокот на податоци. Користењето на облаче во овој случај ги олеснува работите во случај на побарување на голем проток на податоци на повеќе корисници истовремено.

3.2.2 Пресметување во магла

Пресметувањето во магла го проширува пресметувањето во облак и неговите сервиси на крајот на мрежата. Земајќи го тоа во предвид, Cao и останатите во [33] предлагаат FAST, систем кој користи пресметување во магла за анализа и мониторирање на пациенти со преживеан мозочен удар. Авторите имаат развиено сет на алгоритми за откривање на мозочниот удар базирани на техники на филтрирање за да се олесни детекцијата на мозочниот удар. Тие дизајнирале систем за откривање на удар во реално време базиран на пресметување во магла кој прави поделба на задачите за откривање на ударот помеѓу крајниот уред и облакот. Предложениот систем постигнува високо ниво на чувствителност и специфичност при тестирање на податоци од реалниот свет. Во исто време, времето за одговор и потрошувачката на енергија се блиску до најефикасните

постоечки пристапи.

Друга употреба на пресметувањето во магла е претставена од Stantchev и останатите [34]. Тие предлагаат архитектура со три нивоа за паметна здравствена инфраструктура со цел да се обезбеди ефикасна архитектура за апликации за нега на постари лица. Пресметувањето во магла ја подобрува архитектурата обезбедувајќи мали задоцнувања, мобилна поддршка, информација за локацијата и сигурносни мерки.

Пристапот со користење на пресметување во магла може да се смета како прокси сервер, иако целта, употребата и структурата на серверот кој се користи за пресметување во магла не се исти како на класичните прокси сервери. Како што е прикажано во [35] серверите кои се користат за пресметување во магла вообичаено можат да опслужат еден корисник. Серверот е многу помал од оној на облакот и ги чува само податоците од самиот корисник, кои многу лесно може да исчезнат од серверот поради хардверско оштетување, вирус, итн. Доколку серверот исчезне, многу лесно може повторно да биде креиран бидејќи постои копија од истиот на облакот. Овој тип на сервери може да се користат со или без Интернет поврзување бидејќи самиот сервер е подигнат на локалниот компјутер на корисникот. Доколку одредена веб страна ја користи оваа архитектура, истата ќе може да биде достапна и без Интернет поврзување. Веб страната ќе биде копирана на локалниот компјутер и корисникот ќе може да ја користи и кога ќе нема Интернет поврзување. Разликата од ваквата копија и самиот оригинал е во тоа што корисникот ќе може да користи само ограничени содржини. Доколку корисникот направи промена на локалниот сервер додека нема Интернет поврзување, синхронизацијата со облакот нема да се направи веднаш, туку кога ќе се поврзе локалниот компјутер на Интернет.

3.2.3 Пресметување во роса

Иако моментално како сервери за процесирање во роса може да се користат десктоп и лаптоп компјутери и некои помоќни таблети, предвидувањата се дека во блиска иднина голем дел и од паметните телефони ќе може да ја преземат таа улога. Доколку ваквата архитектура се користи при мобилно пресметување, корисникот може да дозволи неговиот мобилен уред да се користи како сервер за процесирање на кој ќе пристапуваат други корисници во близина. Доколку сите корисници дозволат нивниот мобилен уред да се користи како сервер, тие би ги зголемиле перформансите на мобилниот уред и би ги прошириле функционалностите кои можат да ги користат на својот мобилен уред, дури и без Интернет поврзување.

3.3 Истражувања од областа на темата

Истражувањата поврзани со искористеноста на OCR алгоритмот и пресметувањето во облаче, облак, магла или роса се сè поинтензивни во последно време. Истражувањата одат во насока на подобрување на тоа да се подобри точноста на алгоритмите за пре-

познавање карактери намалување на времето за процесирање на барањата од самиот корисникот. Исто така голем дел од истражувачите вложуваат големи напори во тоа да се намали користењето на ресурсите на самиот мобилен уред и извршувањата да се прават на оддалечен сервер, а во исто време што е можно побрзо и поефикасно да се извршат истите.

Неколку научно-истражувачки проекти се обидоа со користење на камерата на мобилниот телефон да развијат интересни апликации кои би користеле OCR алгоритам. Повеќето од нив се развиени да користат сервери во позадина за процесирање на сликите без употреба на мобилниот уред. Во [36] се презентирани мобилни апликации за туристи кои лесно го препознаваат текстот на патокази, инфотабли, банери, книги, на земјата во која се наоѓаат, го претвораат текстот од сликата и го преведуваат на мајчиниот јазик на туристот.

Очилата развиени од Google се моментално најпопуларни овозможувајќи анализа на сликата и визуелна пребарувачка алатка развиена за Андроид и iOS уреди. Апликацијата е способна да препознае и преведе текст од различни знаменитости, бар-кодови, итн. Тие функционираат со сликање на слика на самиот уред и потоа испраќање на истата преку Интернет за препознавање до базите на податоци во компанијата.

World Lens [37] е апликација за iPhone мобилниот уред која преведува видлив текст и ги прикажува резултатите на екранот во реално време. Системот функционира без користење на Интернет поврзување и моментално овозможува само преведување од англиски на шпански и обратно. На академско ниво Fragoso и останатите [38] развиле апликација која користи веб базиран сервис за преведување.

Динамичен, енергетски ефикасен модел базиран на пресметување во облаче е претставен во [39]. Тој се базира на решавање на проблемот со дополнителната побарувачка на енергија за време на безжичната комуникација преку проширување на динамичен модел базиран на пресметување во облаче. Базирајќи се на барањата на индустријата, овој модел нуди уникатен механизам за избегнување на потрошувачката на енергија кога корисникот се наоѓа во нестабилна мрежна околина. Моделот е еден вид на веб сервис кој се фокусира на ефикасна комуникација помеѓу уредите на корисниците и серверите на облачето.

Подобрувајќи ја идејата за софтверско дефинирано мрежно поврзување, Cui и останатите [40] предложиле софтверски дефиниран кооперативен модел за растоварување на пресметувањето (Software Defined Cooperative Offloading Model(SDCOM)), каде соодветен контролер е распореден на податочниот мрежен портал и ги закажува извршувањата на централизиран начин сè со цел да ја зачува батеријата на мобилниот уред. Мобилните уреди ги прават пресметувањата локално, соработуваат со другите уреди во близина или во координација со контролерот ги префрлуваат на облачето.

Апликациите за виртуелна реалност се многу нетолерантни на задоцнување па дури и мали доцнења може да го уништат корисничкото искуство. Оттука пресметувањето во магла има потенцијал да постане главен играч во доменот на виртуалната реалност. Зао и останатите [41] направиле компјутерска интерактивна игра базирана на виртуална

реалност која користи пресметување во магла. Кога корисникот ја игра играта, сурови податоци собирани од EEG сензори се генерираат и класифицираат за да ја откријат состојбата на мозокот на играчот. Ваквите податоци се едни од најтешките задачи за пресметување бидејќи мора да се извршат во реално време. Системот ги користи и двата сервери за пресметување во магла и во облак, комбинација која му овозможува на системот континуирани и во реално време класификации на состојбата на мозокот.

Во докторската дисертација се обработени сите сценарија кои што би овозможиле подобро и побрзо пресметување. Покрај пристапот со користење на мобилниот уред за пресметување, опфатени се и пристапите со користење на облак, облаче и пресметувањето во магла. Со тоа се зголемува опсегот на пристапи кои што можат да се користат за пресметување на мобилни апликации кои побаруваат големи ресурси.

Главната разлика помеѓу решенијата споменати погоре и решението развиено во докторската дисертација е начинот на којшто се прават процесирањето на сликата и OCR процесирањето. Решението во докторската дисертација е независно и може да се постави на мобилен уред, облаче или облак и може да го користи само мобилниот уред, само облачето или само облакот за процесирање.

Решението претставено во докторската дисертација во споредба со погоре споменатите пристапи има многу подобро време за процесирање и препознавање на карактерите на самиот мобилен уред, и поголема точноста при препознавањето, независно од типот на сликата и бројот на карактери на истата.

Глава 4

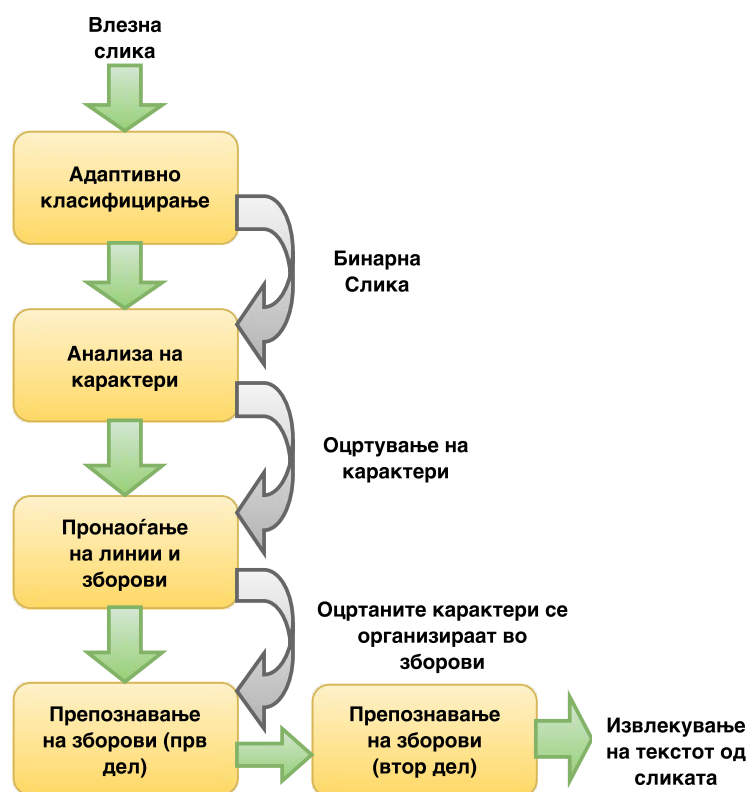
Решенија за OCR со мобилен уред

4.1 OCR процесирање

По долга анализа на OCR алгоритми, се одлучив за Tesseract алгоритмот, како бесплатно решение кое што може да се користи на различни оперативни системи.

4.1.1 Архитектура на Tesseract

Patel и останатите го објаснуваат Tesseract OCR процесирањето чекор по чекор како што е презентирano на Слика 4.1



Слика 4.1: Tesseract OCR процесирање

Првиот чекор во процесирањето е адаптивно класифицирање што ја претвора сликата во бинарен формат. Потоа се прави анализа на сите карактери и се прави нивно оцртување. Ова е многу корисен чекор бидејќи OCR има најголема точност на црно-бели слики. Карактерите кои се предходно оцртани се претвораат во региони кои потоа се организираат во линии. Текстуалните линии и региони се анализираат, се прави фиксирање на одредена област и изедначување на фонтоот, за потоа да се направи поделба по зборови.

Препознавањето се одвива во две фази. Во првата фаза секој збор во текстот се препознава, а потоа следи адаптивно класифицирање. Адаптивното класифицирање се обидува да го препознае текстот на многу попрецизен начин. Во втората, финална фаза, се прави препознавање на секој карактер и со тоа се завршува процесот на препознавање.

Процесот на OCR препознавање се состои од неколку чекори: оптичко скенирање, прет-процесирање, сегментација, извлекување на карактерите, препознавање на карактерите и пост-процесирање.

4.1.2 Оптичко скенирање

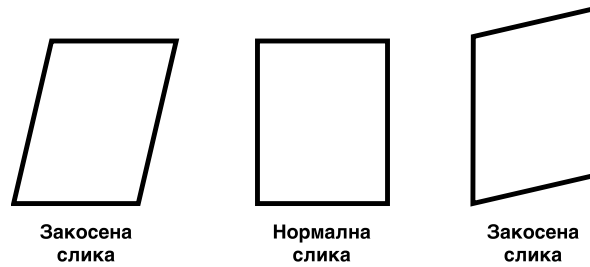
Првиот чекор кај OCR процесирањето е оптичко скенирање што значи, скенерот го чита испечатениот текст и го преведува во формат разбирлив за компјутерот. Процесот се базира на претворање на сликата во дигитален формат со дефинирање на граничен простор со вредност 1 или 0, каде што 1 значи дека просторот е исполнет додека 0 дека просторот е празен. Овие вредности формираат битмапирачка матрица што се презентира на екранот и сликата се снима во соодветен формат како BMP или TIFF.

4.1.3 Детектирање на косина и прет-процесирање

Прет-процесирањето содржи неколку чекори кои ја подготвуваат сликата за сегментирање. Потенцијалното искосување на сликата може да предизвика голем проблем во OCR процесирањето. Детектирање на искосувањето, како прв чекор во претпроцесирањето на сликата, потврдува дека сликата е во правилна насока, а не со косина кон една или друга страна. Процесот на детектирање на косината подразбира едноставна ротација на сликата, до точка каде што хоризонталните и вертикалните линии на сликата се целосно паралелни на хоризонталната и вертикалната оска соодветно како што е прикажано на Слика 4.2 [42].

4.1.4 Бинарно прет-процесирање

Процесот на бинаризирање се состои од два подчекори: претворање на предходно скенираната слика во сива слика, и потоа таа слика во бинарна слика на која се идентификувани белите и црните делови. Потенцијалните проблеми што може да се појават



Слика 4.2: Детектирање на косина на сликата

се претставени на Слика 4.3 каде може да се види дека при процесот на бинаризација делови од карактерите фалат или не се доволно јасни.



Слика 4.3: Бинаризација на сликата

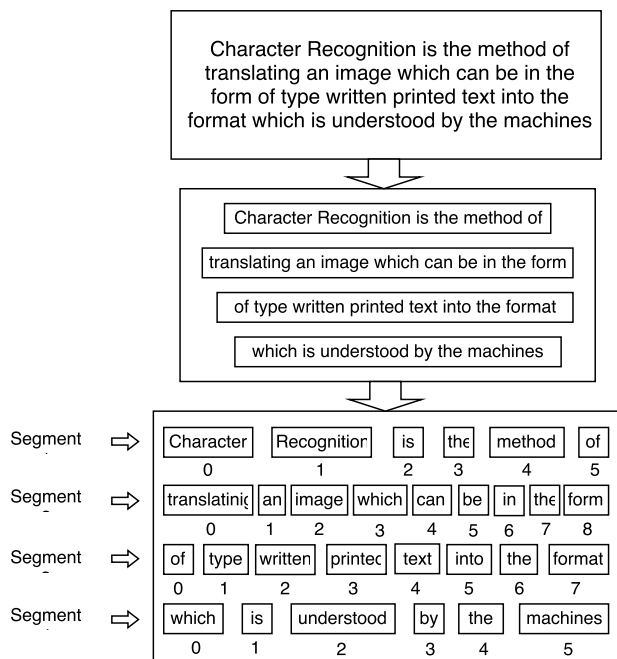
Најдобар начин за бинаризација на сликата е да се одреди гранична вредност и потоа да се класифицираат сите пиксели со вредности под дадената гранична вредност како "Бела" додека сите пиксели над дадената гранична вредност како "Црна" [42].

4.1.5 Сегментација

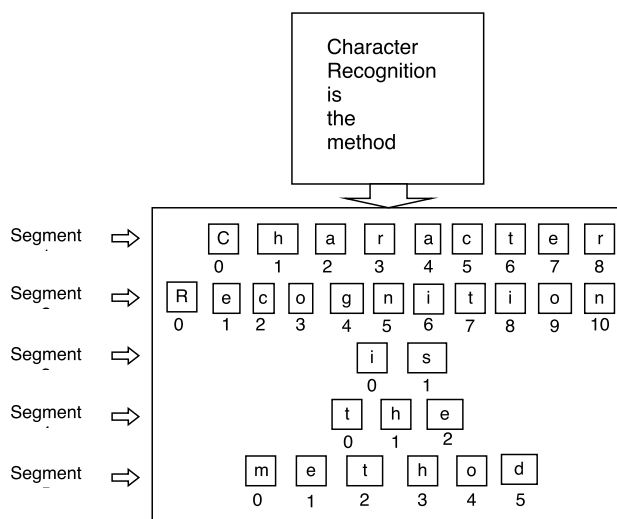
Сегментирањето е процес на издвојување на карактерите или зборовите од текстот. За време на сегментирањето потребно е да се лоцираат регионите каде се наоѓаат податоците и да се одделат од останатите делови на сликата [43].

Процесот на сегментирање започнува со сегментирање на линиите, што подразбира сегментирање на карактерите и позиционирање на зборовите во низа. Пример за ова е прикажан на Слика 4.4 каде што параграфот е поделен во линии и секоја линија е поделена во различни зборови. Следниот чекор е сегментирање на карактерите, каде секој карактер посебно се сегментира и се чува во низа како што се гледа на Слика 4.5.

Оваа техника лесно се имплементира но проблемот настанува кога карактерите се допираат еден со друг или пак карактерите се поделени и се состојат од неколку делови.



Слика 4.4: Сегментација на карактерите во линија

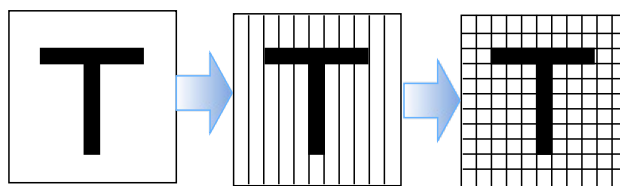


Слика 4.5: Поединечна сегментација на карактерите

4.1.6 Извлекување на карактери

Сликите со карактери содржат различни нумерички карактеристики како што се ширината на карактерите, висината на карактерите како и пикселите во различни делови на сликата. Процесот на извлекување на карактери е имплементиран на една единствена слика со карактери и започнува со прилагодување на сликата во формат на дводимензионална бинарна матрица како што е претставено на Слика 4.6.

Ова не е тривијален процес бидејќи ширината и висината на карактерите на сликата може да варира и поради тоа се употребува невронска мрежа за да се добие ефикасно мапирање. На самиот крај оваа матрица се конвертира во еднодимензионална низа која

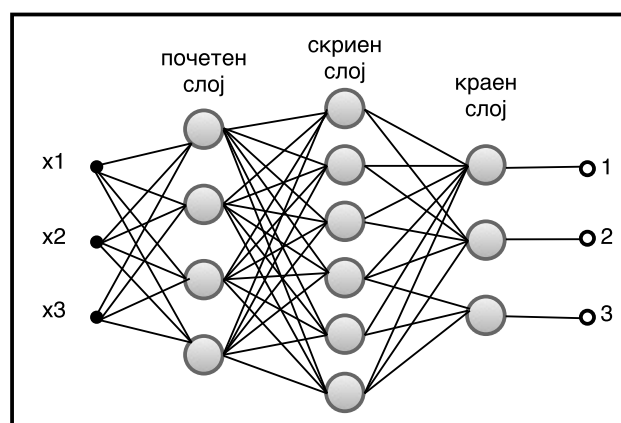


Слика 4.6: Извлекување на карактерите од сликата

потоа се користи за препознавање на карактерите [42].

4.1.7 Препознавање на карактерите и постпроцесирање

Процесот на класификација ги користи карактеристиките извлечени од карактерите на сликата за да го утврди типот, главно од страна на повеќеслојната перцепција, притоа мапирајќи делови од влезните параметри во излезни. Слика 4.7 ги претставува слоевите на повеќе слојна перцепција (почетен, скриен, крајна), каде што почетниот слој е еднодимензионална низа добиена со извлекување на карактерите. Сите преостанати слоеви се меѓусебно поврзани со тоа овозможувајќи му на скриениот слој да го произведе крајниот резултат[42].

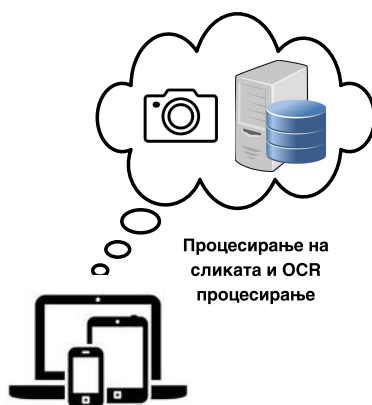


Слика 4.7: Повеќе слојна перцепција

Последниот чекот во OCR процесот е постпроцесирањето кое ги печати препознаените карактери во структурирана форма со пресметување на еквивалентната Unicode вредност во соодветна Unicode табела.

4.2 Различни пристапи за процесирање

Моделирањето претставува дистрибуција на соодветни процесирачки и комуникациски активности во идентификувани ресурси во секое решение. На Слика 4.8 се прикажани основните комуникациски и процесирачки активности за остварување на OCR на дадена слика кога сликата е сликана со мобилен уред и резултатите од процесирањето се претставени на истиот. Сликата е сликана со помош на камера од мобилниот уред.



Слика 4.8: Комуникациски и процесирачки активности за OCR процесирање на слика сликана со мобилен телефон

Истата може да биде процесирана на самиот мобилен уред или пренесена да се процесира на облак или облаче. Резултатите потоа се пренесени назад на мобилниот уред.

Процесирањето на сликата започнува со претворање на сликата во црно-бел формат, двојно зголемување за подобро препознавање и сиво процесирање за детално презентирање на контурите на сликата. Процесот продолжува со *OCR процесирање*. OCR процесирањето опфаќа оптичко скенирање за да се елиминира потенцијалното закосување на сликата, а продолжува со бинаризација, сегментација и извлекување на карактерите пред да заврши со препознавање на карактерите и постпроцесирање.

Спецификацијата каде што соодветна компјутерска компонента ќе биде пронајдена во презентираниот архитектура ја дефинира целосната методологија. Во случајот на процесирање на мобилниот уред, нема потреба од префрлување на сликата на далечен сервер и поради тоа процесирањето е прилично едноставно (Слика 4.8).

Покрај сценариото за процесирање на мобилниот уред, дефинирани се уште три други архитектурни сценарија специфицирајќи ја локацијата на процесирање на сликата и OCR процесирањето, како и анализа на времето на одговор, задоцнувањето како и протокот на податоци. Целта на истражувањето е да се евалуираат добиените времиња на одговор за процесирање на сликата и OCR процесирање и потоа споредба на резултатите за да се утврди кое решение е најдобро.

Разликите кај сите сценарија опфатени во истражувањето како и кои се предностите и недостатоците кај истите се прикажани во продолжение:

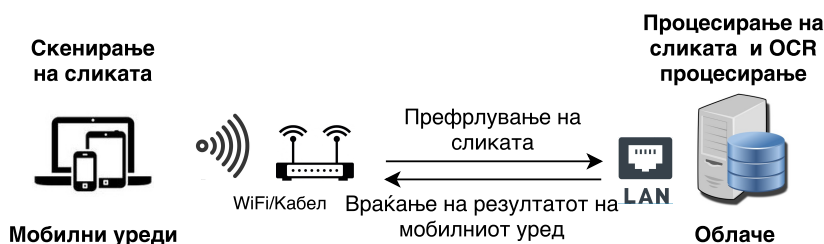
4.2.1 Сценарио М

Сценарио М ги дефинира сите активности што треба да се извршат на мобилниот уред за да се направи процесирање на сликата и OCR процесирање како што е претставено на Слика 4.8. Целото процесирање се прави на мобилниот уред и не се користат други процесирачки единици (сервери).

Мобилниот уред претставува еден вид на сервер на којшто се прави процесирањето без да се користат сервисите на облакот и без Интернет поврзување.

4.2.2 *Сценарио L*

Сценарио L претставува случај каде сите активностите поврзани со процесирање на сликата и OCR процесирање се прават на облачето. Слика 4.9 го претставува протокот на податоци и процесирањето кај ова сценарио. Времето за одговор вклучува пренесување во облачето користејќи Wi-Fi поврзување, процесирање на сликата и OCR процесирање, и враќање на резултатите од процесирањето назад на мобилниот уред.



Слика 4.9: Сценарио L: Процесирање на слика и OCR процесирање во облаче

4.2.3 *Сценарио F*

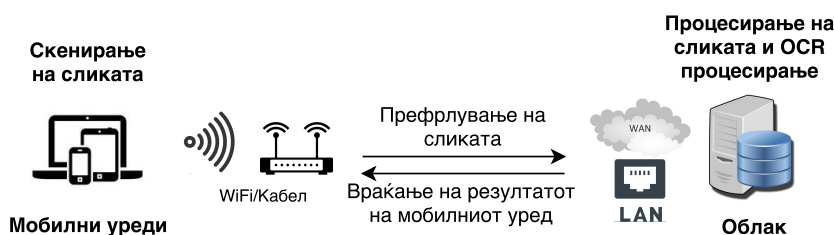
Сценарио F е сценарио каде процесирањето на сликата и OCR процесирањето се прават на сервер хостиран кај мобилен оператор користејќи го пристапот на процесирање во магла. Протокот на податоците и процесирачката архитектура се претставени на Слика 4.10. Времето на одговор вклучува префрлување на сликата на серверот користејќи 3G/4G поврзување, процесирање на сликата и OCR процесирање, и враќање на резултатите назад на мобилниот уред.



Слика 4.10: Сценарио F: Процесирање на слика и OCR процесирање во магла

4.2.4 *Сценарио C*

Сценарио C вклучува процесирање на сликата и OCR процесирање во облак како што е претставено на Слика 4.11. Податоците се префрлуваат на облакот користејќи LAN или WAN поврзување. Мобилниот уред тука служи за сликање на сликата и презентирање на резултатите, додека процесирањето на сликата и OCR процесирањето се прават во облакот.



Слика 4.11: Сценарио С: Процесирање на слика и OCR процесирање во облак

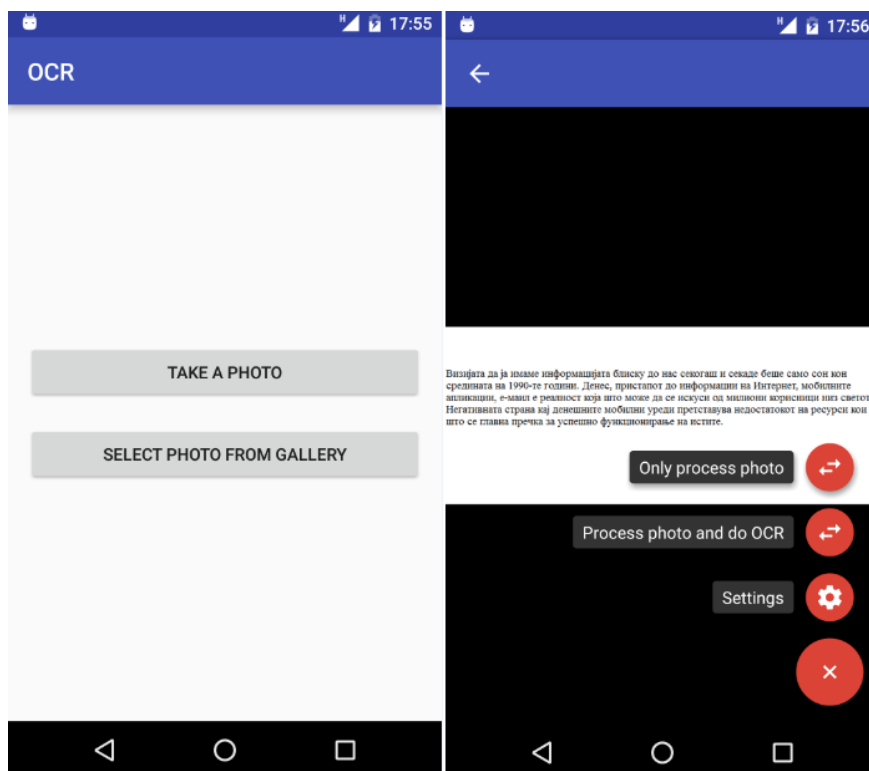
4.3 Опис на решението

Решението претставено во докторската дисертација овозможува избор на различни архитектурни пристапи. Користени се пристапите за пресметување во облаче, облак, магла и пристапот за пресметување на самиот мобилен уред.

Слика 4.12 го претставува GUI интерфејсот на Андроид апликација која е развиена специјално за потребите на ова истражување. Апликацијата има едноставен дизајн и се состои од две копчиња од кои, едно за опцијата да се направи слика со камерата на самиот мобилен уред и друго за преземање на предходно сликана фотографија од мобилниот уред. Кога се селектира опцијата се појавува фотографијата и може да се одбере што да се прави, дали да се прави само процесирање на сликата (Only process photo) или процесирање на сликата и препознавање на карактерите истовремено (Process photo and do OCR). Откако ќе се одбере соодветната опција, резултатот од препознавањето на карактерите се појавува на екранот на мобилниот уред.

Во позадина се покренати сервери кои служат за процесирање на сликата, на кои е инсталиран софтвер којшто е реализиран преку OCR веб сервис. Тука се наоѓаат Tesseract алгоритмот и Leptonica библиотеката потребни за процесирање на сликата и препознавање на карактери. На самите сервери е сместена предходно тренирана датотека која содржи букви, знаци, бројки, специјални карактери, итн. Серверот на којшто се извршува процесирањето на сликата и препознавањето на карактерите се менува во позадина во зависност од тоа за кој пристап станува збор. Со тоа се овозможува детална анализа и евалуација на резултатите за тоа кој од пристапите е најдобар.

Се користат неколку архитектурни решенија за да се направи префрлување на извршувањето надвор од мобилниот уред и тоа, пристап за пресметување во облаче кое се наоѓа во близина на мобилниот уред и пристап со користење на облак сместен во Орегон(USA). Поврзувањето со овие сервери се прави преку Wi-Fi, 3G/4G, LAN или WAN конекција во зависност од тоа кој пристап се користи. Ова дава можност да се опфатат сите архитектурни пристапи обработни во докторската дисертација.



Слика 4.12: OCR Апликација

Глава 5

Методологија за експериментално истражување

Направено е и експериментално истражување за да се провери кое архитектурно решение, пресметување во облаче, пресметување во облак, пресметување во магла или пресметување на мобилниот уред е најдобро. Претставена е методологијата на истражување за да ги дефинира деталите за тоа како се изведени експериментите и евалуацијата на резултатите. Различни тестни сценарија и решенија се дефинирани и споредени. Тестните сценаријата ги дефинираат параметрите користени при тестирањето за да се оценат различните зависности.

5.1 Тестна околина

За тестните сценарија се користи трослојна архитектура која се состои од мобилен уред, облаче и облак. Најнискиот слој се состои од мобилен уред, следен е средниот слој (облачето) кој се состои од сервер лоциран во близина на мобилниот уред и највисокиот слој е податочниот центар сместен на облакот. Комуникацијата помеѓу најнискиот и средниот слој се врши преку Wi-Fi, Bluetooth, 3G/4G или друго радио поврзување. Комуникацијата помеѓу средниот и највисокиот слој е преку WAN поврзување.

Табела 5.1 ги претставува соодветните хардверски и софтверски карактеристики на пресметувачките компоненти кои се користат во презентираната архитектура.

Четири различни околинати се користат во експериментот, обележани со M, C, L и F соодветно за решение со користење само на мобилниот уред за процесирање, решение со користење на облак како сервер за процесирање, решение со користење на облаче за процесирање и решение за процесирање во магла. Мобилното решение го користи само мобилниот уред како процесирачка единица, додека кај останатите процесирањето се извршува на одреден сервер. Кај решението базирано на процесирање во облаче се користи мал сервер сместен во близина на мобилниот уред и пренесувањето на податоците се прави преку Wi-Fi поврзување или кабел. Решението базирано на пресметување во магла користи 3G/4G поврзување за испраќање на податоците кон сервер

Табела 5.1: Техничка спецификација на тестната околина

	Мобилен уред	Облаче/Магла	Облак
Инфраструктура	LG Nexus 5	HP Probook 450 G2	Amazon EC2 Cloud
Оперативен систем	Android 6.0.1	64-bit Windows 8.1 Enterprise	64-bit Windows Server 2012
Процесор	Quad-core Krait 400	Intel(R) Core(TM) i5-4210U	Intel(R) Xeon(R) CPU E5-2670 v2
Фреквенција на процесорот	2.3GHz	2.4Ghz	2.5Ghz
РАМ меморија	2 GB	4GB	1GB

лоциран во близина на базната станица на мобилниот оператор. Кај решението кое се базира на пресметување во облак, мобилниот уред комуницира директно со серверот на облакот користејќи WAN поврзување за пренесување на податоците од мобилниот уред.

5.2 Прилагодувања за експериментот

Многу софтверски алатки беа прецизно конфигурирани за да се постигнат целите на истражувањето. Имплементиран беше Tesseract OCR алгоритмот заедно со Лептоника библиотеката (отворен код која содржи софтвер кој е широко корисен за обработка и анализа на слики) за процесирање на слики која што може да препознава широк спектар на формати на слики и да ги претвора истите во текст на повеќе од 60 јазици. Лептоника библиотеката за препознавање на слики беше инсталирана на облачето и облакот.

Користени беа следните почетни обработки на слики:

- *Црно и бело* процесирање за претворање на сликата во црно бел формат.
- *Двојно зголемување* на сликата за полесно препознавање.
- *Сива обработка* на сликите за добивање на подетални линии

Овие обработки се изведуваат секвенцијално една по друга.

За да се изведе OCR процесирањето, листа на специјални карактери од предефинираниот јазик мора да се достават на Tesseract системот. Во мојот експеримент, Tesseract алгоритмот беше трениран да препознава карактери со кирилична поддршка на македонски јазик. За таа намена, 100 карактери беа користени како основа за тренирање, со што Tesseract алгоритмот беше доволно истрениран за да препознава карактери кога тие ќе се појават во селектираниот текст од сликата.

Тренингот на алгоритмот се изведува со помош на документ со зборови, реченици, симболи и броеви од предефинираниот јазик, кој содржи задолжителни 10-20 примери од секој карактер од јазикот [44]. Како најсоодветна тренинг Tesseract апликација за експериментите беше користена Serak тренинг апликацијата.

Тренирањето на фајловите беше многу специфично поради карактерите со кирилична поддршка, па потребно беше да се направат дополнителни рачни корекции на грешките добиени од Tesseract за време на тренингот. Со овој процес успеав да стигнам до 70% успешност во препознавањето на карактерите.

5.3 Тестни сценарија

Различните сценарија ги дефинираат различните процесирачки архитектури и влезни параметри за тестните случаи. Тие главно целат кон тестирање на нивната зависноста од волуменот и комплексноста на процесирањето, како што се различните фонтови и други различни пречки, кои го спречуваат квалитетното процесирање на сликите. Секое тест сценарио претставува скенирана слика од текст сликана со мобилен уред и може да има многу недостатоци како што се, закосување, слаб контраст, заматување на сликата, итн.

Поради тоа се дефинирани неколку тестни сценарија кои ги содржат следните критериуми:

- *Волумен* претставен преку бројот на препознаени карактери.
- *Различности* дефинирана преку бројот на фонтови, големината на карактерите, мешавината на знаци и други елементи на сликата.
- *Комплексности* на сликата дефинирана преку квалитетот на сликата

Различниот волумен се специфицира преку тестни сценарија кои се базираат на бројот на карактери (N_C) скенирани на сликата. Спектарот на *микро волумен* се дефинира со една линија текст од максимум 100 карактери. Спектарот на *мал волумен* се дефинира со параграф со текст од максимум 1000 карактери што одговара на една четвртина од страна A4 формат. Спектарот со *среден* и *голем волумен* се состои од цела страна со текст од околу 4000 или 10000 карактери соодветно. Табела 5.2 ги дава деталите за основниот број на карактери искористен за секое тестно сценарио посебно.

Табела 5.2: Број и големина на карактери користени во тестните сценарија

волумен	големина на текст	број на карактери
<i>микро</i>	линија	50, 75, 100
<i>мал</i>	параграф	200, 400, 600, 800
<i>среден</i>	пола страна	1000, 1200, 1400, 1600, 2000, 2500, 3000, 4000
<i>голем</i>	цела страна	5000, 6000, 8000, 10000

Во мојот експеримент има слики со големина од 5, 10, 20, 30, 40, 50, 100, 150, 200, 250, 300, 350, 400, 450, и 500 KB. Поедноставена презентација се користи за три типови на слики:

- 5KB за *мали слики*,
- 50KB за *средни слики* и

- 500КВ за големи слики.

Комплексноста на OCR процесирањето зависи од користењето на различните фонтови, број на знаци и специјални карактери на сликата, различни големини и стилови на карактери (закосен, задебелен, подвлечен), користење мали и големи букви, итн. За да се направи разлика помеѓу сликите, се дефинира индикатор на различност со две вредности, 1 и 2. Кога сликата содржи карактери со ист фонт и иста големина на фонт тогаш процесирањето е многу едноставно и индикаторот на различност е 1. Меѓутоа, ако сликата содржи карактери со различна големина на фонт, различен стил, и голем број карактери тогаш индикаторот за различност е 2. Во овај случај процесирањето е многу комплексно и може да влијае на процесирачкото време и точноста. Овие два типа индикатори на различност се претставени на Слика 5.1

Recently with the increasing availability of digital cameras and mobile device, increased interest for camera based OCR appeared. Demanding of OCR mobile based applications which would enable people to capture, recognize and translate different languages is huge. They can enable people to understand any native country language and also seamlessly fire a travel specific search to get info regarding the country like museums, restaurants, hotels, culture, temples, books, movies, songs, etc [1].

OCR applications are already used on desktop computers. Images have to be uploaded manually or should be captured directly using the PCs webcam. The image quality of the standard desktop webcam is only 1.3 megapixels, so the quality of OCR may not be satisfactory. Therefore the accuracy of OCR on images captured using smartphones would be better than those captured using webcams. However, image processing on mobile phones is a new field with many challenges due to limited hardware, limited CPU utilization, image quality and connectivity. Phones with high quality cameras, powerful CPUs, and memory storage devices are becoming increasingly common. Most of the existing OCR technology for smart phones are likely to be remotely executed by the servers which do not face limited memory and limited processing power challenge moreover they suffer from networking delay [1]. Our challenge here is to find a solution and use both, mobile device and server as processing units to run OCR and produce good results.

Aleksandar Bahtovski
Ss. Cyril and Methodius University
Faculty of Information Sciences and Computer Engineering
Skopje, Macedonia
Email: abahtovski@gmail.com

Abstract—Mobile devices have evolved to powerful handheld devices in the last few years. Today most of the mobile devices are capable of capturing images, recording videos and browsing. The openness of operating systems led to development of new applications and markets and enabled greater integration. Existing social applications, like business card readers, sign detectors, translation applications, helps people to quickly gather the information in digital format. Among those applications are also applications which are using OCR technology which is a powerful tool for bringing information from analog to digital world. The usage of this technology can be found in creating digital libraries, recognizing text from images, recognizing hand written documents, translation, etc. There are many open source OCR engines for desktop platform and only few available for mobile devices. In this paper we are presenting our work with OCR technology by using mobile cloud-cloudlet architecture. We present a mobile-cloudlet-cloud architecture, created as a solution to support applications that need massive parallel operations, such as OCR. The conducted experiments are conducted on mobile, cloud and mixed cloud-cloudlet processing. The evaluation confirms which approach performs better, analyzing the mobile phone resources and processing time.

Index Terms—Mobile Cloud Computing, Optical Character Recognition, Tesseract, Cloudlet

Слика 5.1: Примери на различност а) Различност 1, б) Различност 2

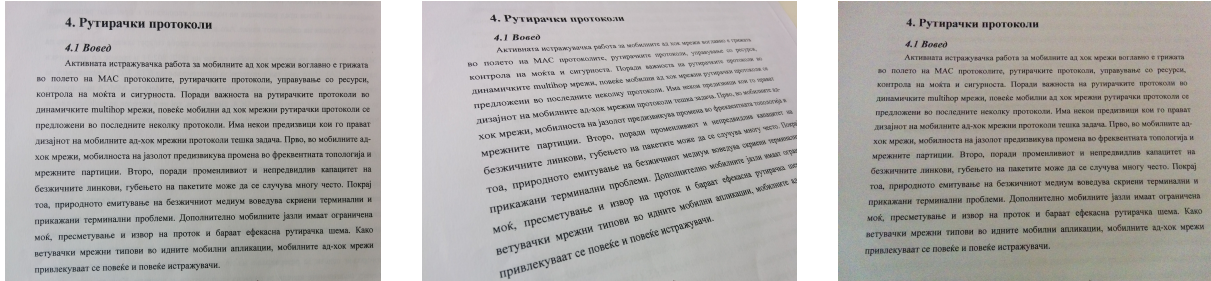
Различност 1 ги опфаќа случаите кога се користи ист фонт и иста големина на фонт. *Различност 2* ги опфаќа случаите каде сликата содржи карактери со различен фонт, различна големина на фонт, различен стил и голем број на карактери.

Комплексноста на сликите може значително да влијае на процесирањето и исто така да ја намали точноста во препознавањето на карактерите. Тестните сценаријата опфаќаат три типови на комплексност на сликите:

- *Нормална* ги опфаќа сценаријата кога сликата е снимана под прав агол, задржувајќи ги хоризонталните и вертикалните аспекти на сликата;
- *Закосена* ги опфаќа сликите снимани под одреден агол без запазување на вертикалните и хоризонталните аспекти и со тоа е нарушена сликата пред да се прави процесирање на истата и препознавање на карактерите;

- *Замашена* ги опфаќа сценаријата со слаба светлина, слаб контраст, сенки на сликата, итн.;

Слика 5.2 ги претставува трите типови на комплексност на сликата.



Слика 5.2: Комплексност на сликата а) нормална, б) закосена, в) заматена

5.4 Тестни податоци

Секое тестно сценарио опфаќа специфична вредност за волумен, различност и комплексност на сликата. Времето на одговор се пресметува за секоја анализирана активност за време на извршување на тестното сценарио. Секој тест се повторува 5 пати за секое од тестните сценарија и се пресметани просечните вредности од сите измерени времиња на одговор.

За да се споредат различните сценарија се пресметува забрзувањето дефинирано како разлика помеѓу соодветните пресметани времиња. (5.1) го дефинира забрзувањето S_{ML} помеѓу сценариото кога се користи мобилен уред и сценариото кога се користи облаче за процесирање. S_{MF} го дефинира забрзувањето помеѓу сценариото кога се користи мобилен уред за процесирање и кога се користи магла за процесирање, додека S_{MC} сценариото го дефинира забрзувањето помеѓу сценариото кога се користи мобилен уред за процесирање и кога се користи облак за процесирање. Исто така се пресметува разликата во забрзувањето помеѓу сценаријата кога се користи облаче и магла за процесирање S_{LF} и сценаријата кога се користи облаче и облак за процесирање S_{LC} . S_{FC} го пресметува забрзувањето кога се користи магла и облак за процесирање.

$$S_{ML} = \frac{T_M}{T_L} \quad S_{MF} = \frac{T_M}{T_F} \quad S_{MC} = \frac{T_M}{T_C} \quad S_{LF} = \frac{T_L}{T_F} \quad S_{LC} = \frac{T_L}{T_C} \quad S_{FC} = \frac{T_F}{T_C} \quad (5.1)$$

Колку е поголемо забрзувањето толку се подобри перформансите на анализираното сценарио во споредба со останатите.

За да се спореди различниот тип на комплексност се пресметува забрзувањето како сооднос помеѓу времето на одговор при препознавањето на карактери на закосени/заматени слики и слики со Нормална комплексност. (5.2) го дефинира забрзувањето S_{skew} помеѓу просечните времиња на одговор кога се користат закосени слики T_{skew}

и просечните времиња на одговор кога се користат слики со нормална комплексност T_{normal} , како и забрзувањето S_{blur} помеѓу времињата на одговор на заматени слики T_{blur} и нормални слики T_{normal} .

$$S_{skew} = \frac{T_{skew}}{T_{normal}} \quad S_{blur} = \frac{T_{blur}}{T_{normal}} \quad (5.2)$$

Дополнително, забрзувањето на различноста $S_{Variety}$ се пресметува преку (5.3) како сооднос помеѓу процесирачките времиња користејќи различни фонтови претходно дефинирани како различност 1 и различност 2. Треба да се забележи дека ова забрзување се пресметува за секое од тестните сценарија, М, С, L, F како и за сите комплексности на слики, закосена, заматена и нормална, со што се добиваат вкупно 12 тестни случаи за секој од тестните сценарија.

$$S_{Variety} = \frac{T_{Variety2}}{T_{Variety1}} \quad (5.3)$$

Точноста е од голема важност во OCR процесирањето. Таа е пресметана како сооднос помеѓу бројот на точно препознати карактери N_C од сите карактери N_I присутни на сликата (5.4).

$$A = \frac{N_C}{N_I} \quad (5.4)$$

5.5 Цел на експерименталното истражување

Целта на истражувањето е да се провери кој од наведените архитектурни пристапи е најдобар. Преку моделирање на однесувањето на перформансите се овозможува да се провери колкаво е задоцнувањето во комуникацијата при процесирањето и како влијае комплексноста на сликите и различноста на фонтовите. Исто така ова истражување дава одговор на поставените хипотези, односно дали примената на пристапот со користење на облачето како архитектура за процесирање дава најдобри резултати. Споредбата со теоретската анализа дава подетални резултати и овозможува полесно докажување на очекуваните резултати.

Глава 6

Евалуација на експерименталните резултати

Експериментот се состои од тестни случаи дефинирани за сите четири сценарија со два типови на различности и три типови на комплексност на сликата. За секој тестен случај направени се по пет обиди и средните вредности за процесирање на сликата и OCR процесирање се земени во предвид.

6.1 Перформанси за процесирање

За да се пресмета вкупното процесирачко време направен е експеримент со кој се пресметува времето на процесирање на сликата и препознавањето на карактерите на истата користејќи различни големини на слики (IS) и број на карактери (N_C). Поради компресија на сликата успеав да соберам колку што е можно повеќе карактери на сликата и со тоа да ги задржам големините на сликите дефинирани предходно (5KB, 50KB, 500KB). Анализирајќи ги резултатите претставени во Табела 6.1, може да се забележи дека бројот на карактери и операции по карактер незначително придонесува во пресметувањето на времето за процесирање на сликата и препознавање на карактерите.

Табелата 6.1 ги претставува времињата на процесирање во милисекунди во зависност од големината на сликата и бројот на карактери. Четири сетови на мерења се претставени за секоја тестна околина L, F, C и M, соодветно за користење на облаче, магла, облак и мобилен уред за процесирање. Може да се забележи дека процесирањето на облачето како сервер постигнува најдобри перформанси, додека процесирањето на мобилниот уред најлоши, како што првично се очекуваше, и поради кое е препорачано префрлување на процесирањето на одреден сервер. Поголемото задоцнување во комуникацијата кај пресметувањето во магла и облак ги рангира со пониски перформанси споредбено со решението каде пресметувањето е во облаче.

Слика 6.1 и Слика 6.2 го претставува вкупното време на процесирање за сите сценарија.

Може да се заклучи дека за поголеми слики потребно е повеќе време за процесирање,

Табела 6.1: Време на процесирање на сликата и препознавање на карактери користејќи различна архитектура

N_C/IS	Облаче			Магла			Облак			Мобилен уред		
	5KB	50KB	500KB	5KB	50KB	500KB	5KB	50KB	500KB	5KB	50KB	500KB
50	345	567	956	705	999	1652	632	834	1624	745	1100	2689
75	375	584	978	707	978	1476	638	831	1599	787	1095	2705
100	324	526	1021	725	968	1489	667	872	1647	725	1054	2731
200	374	598	976	704	974	1524	636	854	1624	764	1063	2685
400	346	576	952	721	956	1478	676	864	1589	781	1068	2698
600	364	586	978	731	952	1420	677	837	1632	768	1069	2714
800	362	594	1031	705	973	1543	615	799	1624	706	1045	2708
1000	349	542	1001	712	967	1567	699	824	1548	762	1026	2719
1200	327	570	1042	719	938	1498	668	861	1604	728	1037	2654
1400	398	581	1089	707	980	1532	676	859	1607	773	1039	2709
1600	326	573	989	728	1010	1520	674	843	1576	728	1048	2718
1800	308	555	999	714	987	1489	668	875	1567	704	1089	2694
2000	356	581	1001	734	968	1467	693	895	1537	754	1093	2684
2500	382	594	1024	725	975	1542	678	827	1621	702	1103	2702
3000	317	567	1006	726	989	1499	675	846	1603	712	1112	2714
4000	367	546	978	727	997	1503	686	787	1584	767	1085	2679
5000	332	558	985	734	958	1476	672	852	1569	735	1087	2699
6000	347	534	1014	741	967	1472	689	850	1599	745	1134	2748
8000	378	587	1013	718	969	1538	681	864	1621	777	1068	2678
10000	329	594	968	726	945	1492	675	875	1593	759	1096	2701

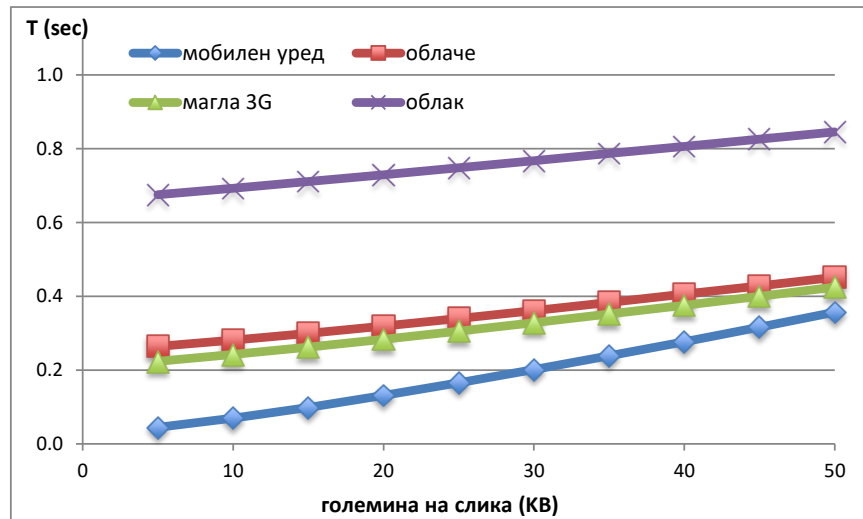
што е во рамки на очекуваното. Исто така може да се види дека постои зависност на процесирачкото време од големината на сликата и бројот на карактери што треба да се препознаат на истата.

6.2 Ефектот на OCR процесирањето

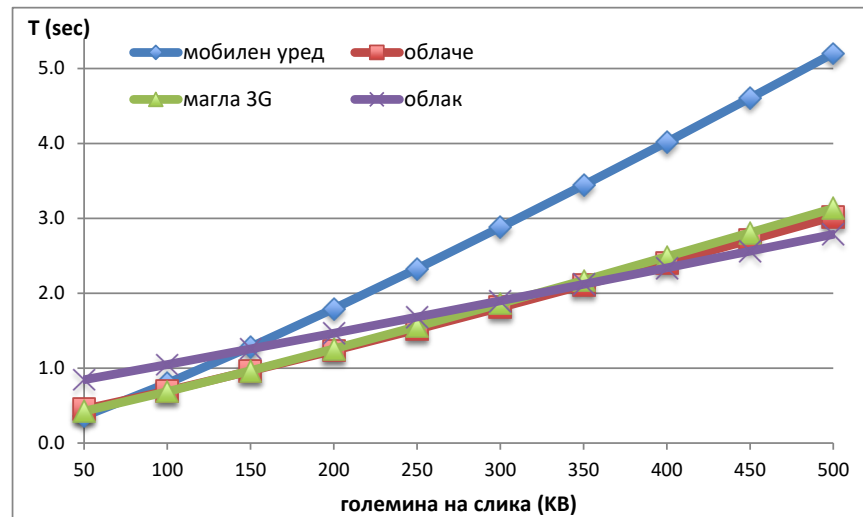
OCR процесирањето игра главна улога во целиот процес на препознавање на карактерите, и поради тоа од големо значење е да се види како влијае врз вкупното процесирање. За да се провери како процесирачката околина влијае врз OCR процесирањето, измерени се OCR процесирачките времиња кај сите четири дефинирани сценарија. Дијаграмите со деталите околу OCR процесирачките времиња се претставени од Слика 6.3 до Слика 6.6. Времињата за процесирање на мерењето се претставени во милисекунди.

6.3 Зависност од комплексноста на сликата

Комплексноста на сликата е еден од клучните фактори при процесирањето на сликата и OCR процесирањето. Вградената камера во мобилниот уред има мала контрола врз условите за осветлување. Нерамномерното осветлување е честа појава поради околината (сенките, рефлесијата) и нееднаквиот одговор при процесирањето од страна на мобилниот уред. Дополнителни компликации настануваат кога се употребува вештачко осветлување, како што е блисот, при што доаѓа до преливање на светлината. Кога ка-



Слика 6.1: Времињата на одговор кај различните сценарија користејќи мали слики

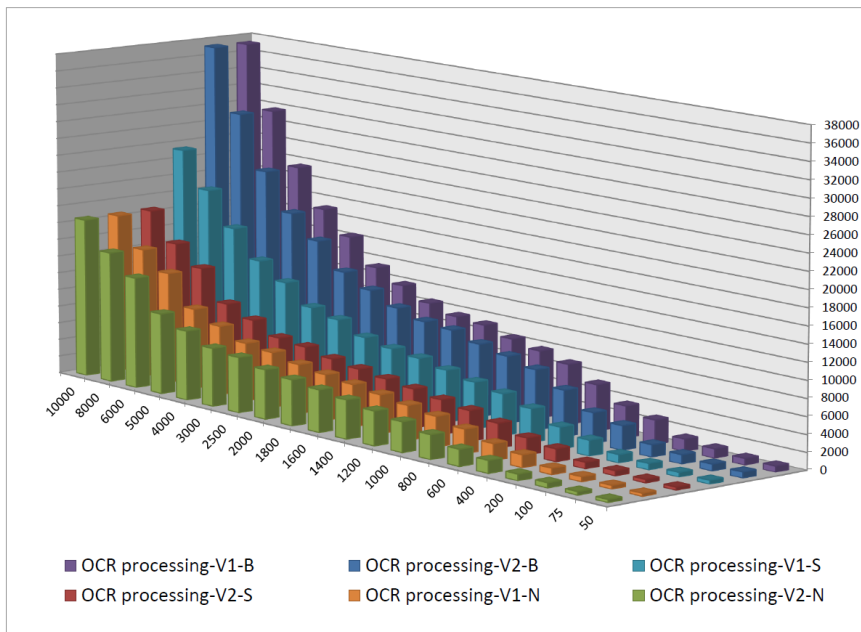


Слика 6.2: Времињата на одговор кај различните сценарија користејќи големи слики

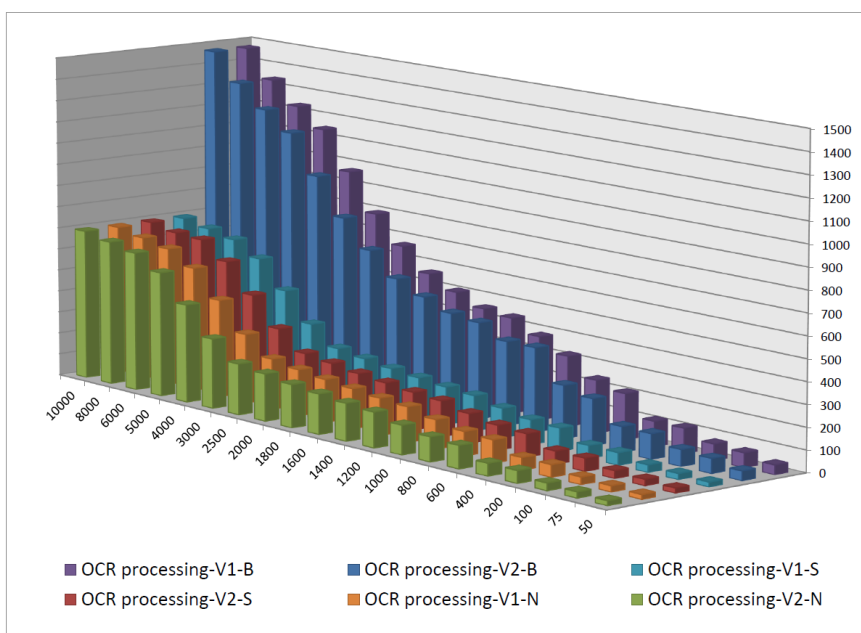
раakterите за препознавање се сликани со видео камера, мобилен уред или друг уред за сликање што не е фиксиран додека се слика сликата, истата може да биде закосена, а со тоа да се добијат многу полоши резултати при препознавањето.

За да се провери како комплексноста на сликата влијае врз препознавањето на сликата и OCR препознавањето, направена е споредба на два случаи, нормална/закосена и нормална/заматена комплексност. Измерено е забрзувањето како однос помеѓу закосени и нормални и заматени и нормални комплексни слики. Како влезни податоци се користени времињата за процесирање кога се користи мобилниот уред за процесирање на сликите. Во некои од случаите процесирачките времиња се скоро идентични. Но кога се користат заматени слики тогаш времето на процесирање е многу поголемо, особено ако се користат слики со голем број на карактери.

Слика 6.7 го претставува забрзувањето S_{skew} помеѓу закосени и нормални комп-



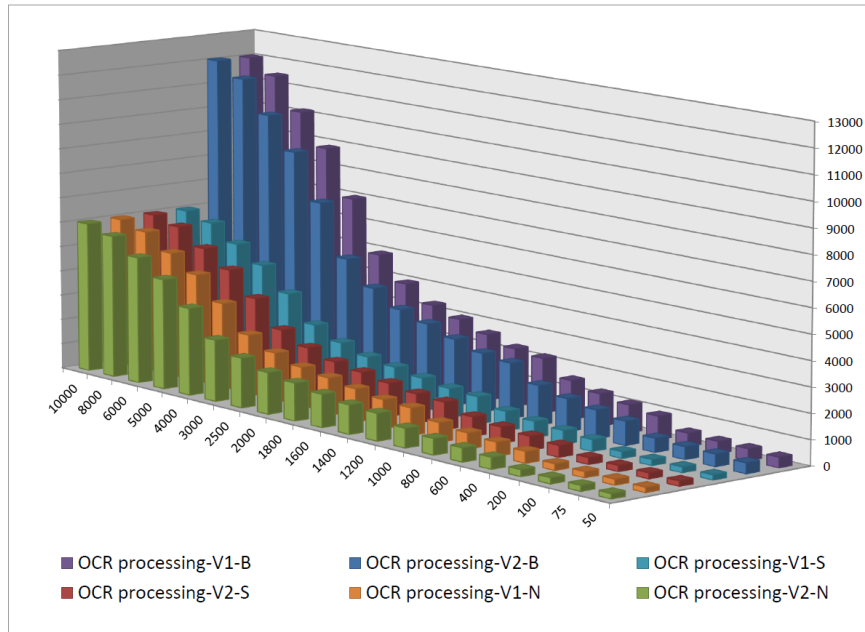
Слика 6.3: Време за OCR процесирање - Мобилен уред



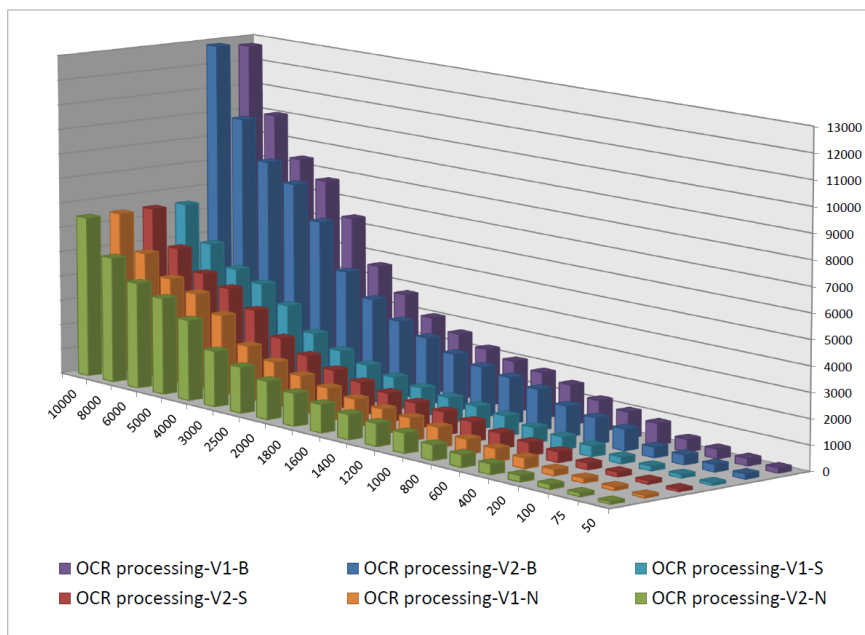
Слика 6.4: Време за OCR процесирање - Облаче

лексни слики додека Слика 6.8 го претставува забрзувањето S_{blur} помеѓу заматени и нормални комплексни слики.

Треба да се забележи дека кај заматените слики е потребно многу повеќе време за процесирање. Анализирајќи ги графициите на забрзувањето, може да се забележи дека процесирачкото време зависи само од големината на сликата, но не и од бројот на карактери на истата. Претставените графици го опфаќаат само сценариото со користење на мобилниот уред, но анализата покажа дека слични резултати се добиваат кога се користат и останатите тестни околин, C (облакот), L (облачето) и F (пресметувањето



Слика 6.5: Време за OCR процесирање - Магла

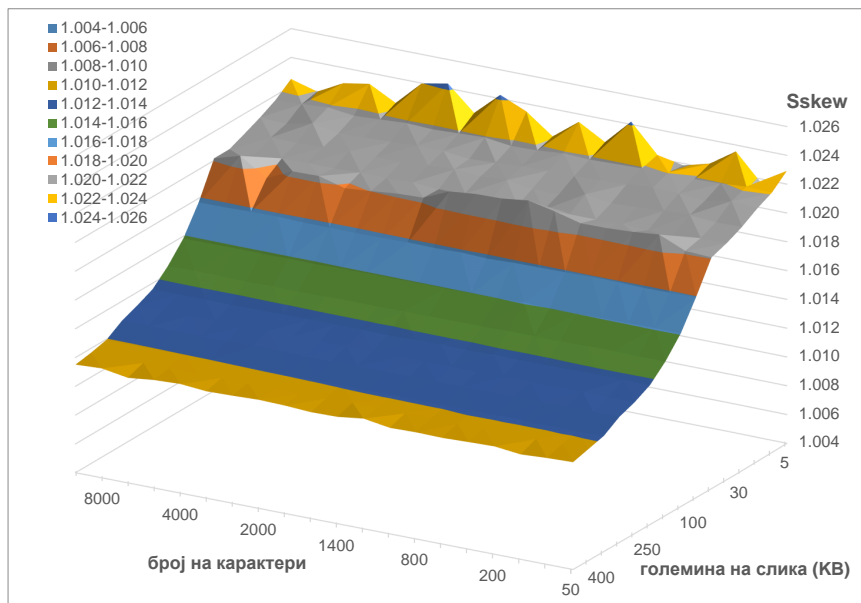


Слика 6.6: Време за OCR процесирање - Облак

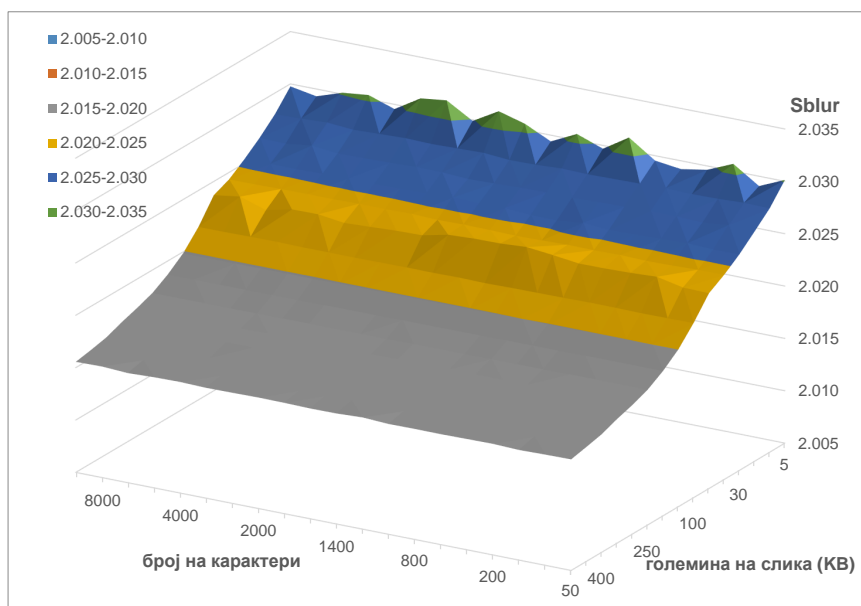
во магла).

Покрај комплексноста на сликата, се прави споредба и на односот од забрзувањето кај различноста на сликите. Сликите 6.9, 6.10 и 6.11 го претставуваат односот од забрзувањето помеѓу Различност 2 и Различност 1 користејќи ги трите типови на комплексност на сликата, нормална, закосена, заматена, соодветно.

Од погоре претставеното се гледа дека закосените слики може да го зголемат времето за процесирање за најмногу 2.4%. Просечното зголемување на процесирачкото време на закосени слики наспроти нормални е 1.67%, со што се заклучува дека времето



Слика 6.7: Забрзување помеѓу Закосена и Нормална комплексност

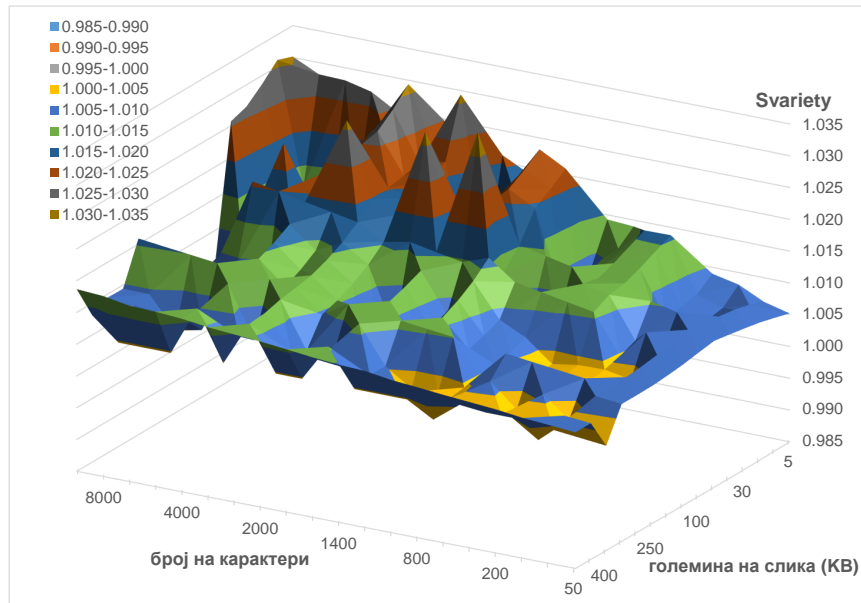


Слика 6.8: Забрзување помеѓу Заматена и Нормална комплексност

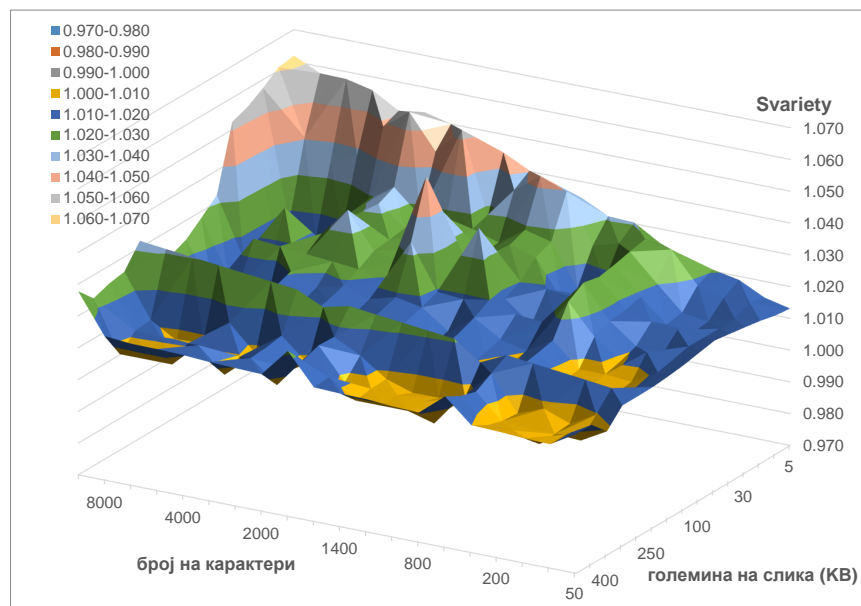
за процесирање на закосени слики е незначително поголемо.

Забележително зголемување на процесирачкото време е добиено кога се користат заматени слики, достигнувајќи просечно зголемување на времето од 102.12%. Во некои од случаите, забрзувањето достигнува вредност од 2.031, што значи дека е потребно 103.17% повеќе време за процесирање.

Како заклучок може да се каже дека, за да нема значително зголемување на времето за процесирање, сликата треба да не е заматена и да има добар контраст. Исто така треба да се внимава кога се фотографира сликата да постои поширок и поблизок извор



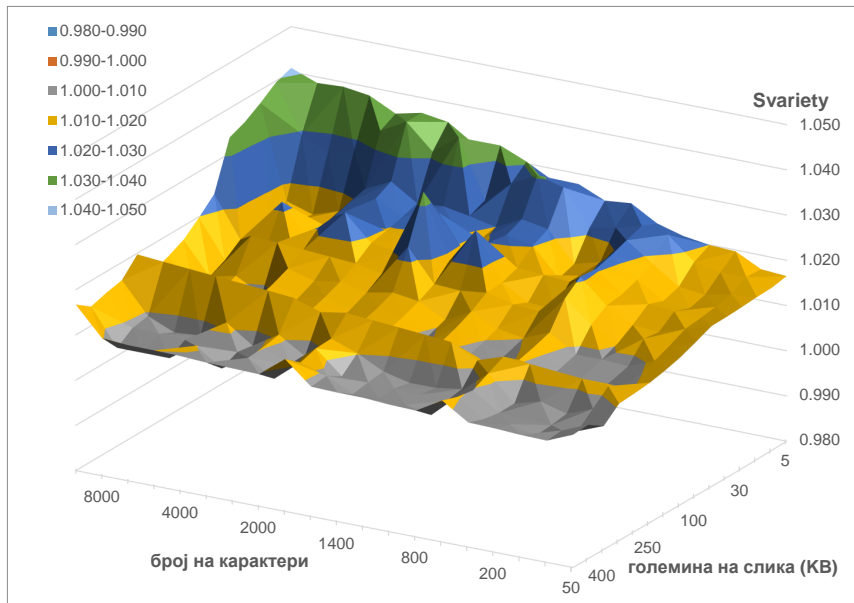
Слика 6.9: Забрзување $S_{variety}$ кај нормална комплексност на сликата



Слика 6.10: Забрзување $S_{variety}$ кај закосена комплексност на сликата

на светлина што ќе го зголеми контрастот и ќе го намали заматувањето. Тоа ќе ја зголеми точноста и ќе го намали времето на процесирање.

Различноста на фонтовите го зголемува времето на процесирање, па поради тоа направена е анализа како различноста на фонтовите влијае врз процесирачките времиња кога се користат трите комплексности на слика (нормална, закосена, заматена). Просечното зголемување на процесирачкото време е за 1.11% кога се процесираат слики со нормална комплексност, 1.51% за заматени слики и 1.91% за закосени слики. Максималните измерени вредности се 3.36%, 6.18% и 4.09%, соодветно за слики со нормална, закосена и заматена комплексност. Од ова може да се заклучи дека различноста на

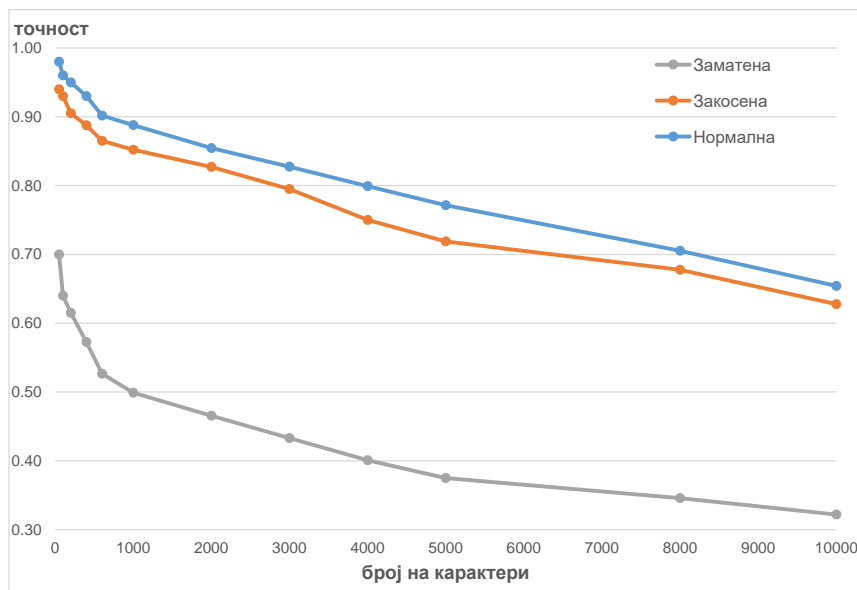


Слика 6.11: Забрзување $S_{variety}$ кај заматена комплексност на сликата

фонтовите незначително ги зголемува процесирачките времиња.

6.4 Точност

За да се провери колку е прецизно препознавањето на карактерите, пресметана е точноста како однос помеѓу бројот на точно препознати карактери и бројот на сите карактери на сликата. Просечните резултати се претставени на Слика 6.12.



Слика 6.12: Распределба на точноста во однос на бројот на карактери на сликата

Податоците претставени на Слика 6.12 се за Различност 1 каде нема мешање на

фонтовите. Покрај Различност 1 направени се и тестови за Различност 2 каде има слично однесување и тренд на резултатите, но помала точност. Табела 6.2 ги содржи минималните, максималните и просечните вредности од споредбата на односот помеѓу точноста кај Различност 1 и Различност 2. Може да се забележи дека максималната разлика е 4.5% додека просечната е помеѓу 1.2% за нормална комплексност на сликите и 2.8% за заматена. Во сите случаи точноста кај Различност 1 е незначително подобра.

Табела 6.2: Однос $Acc_{variety1}/Acc_{variety2}$ за различна комплексност на слики

	Нормална	Закосена	Заматена
MIN	1.002	1.007	1.011
MAX	1.021	1.033	1.045
AVG	1.012	1.020	1.028

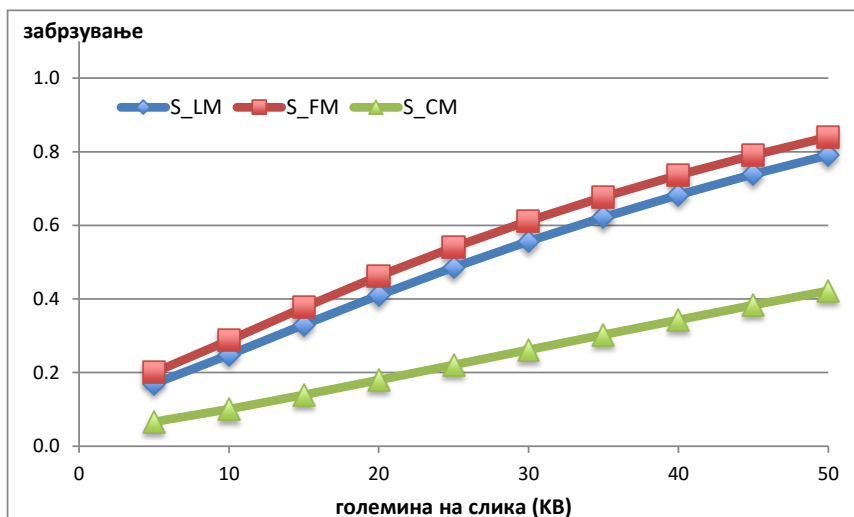
Може да се забележи дека точноста е многу поголема кај слики кои содржат помал број карактери за препознавање. Кај сликите со поголем број карактери за препознавање точноста е многу помала. Закосените слики имаат помала точност, додека заматените слики достигнуваат скоро двојно помала точност од оние со нормална комплексност. Табела 6.3 го претставува односот помеѓу точноста на закосени и нормални слики и односот помеѓу точноста на заматени и нормални слики.

Табела 6.3: Однос помеѓу точноста кај закосени и заматени спроти нормални слики

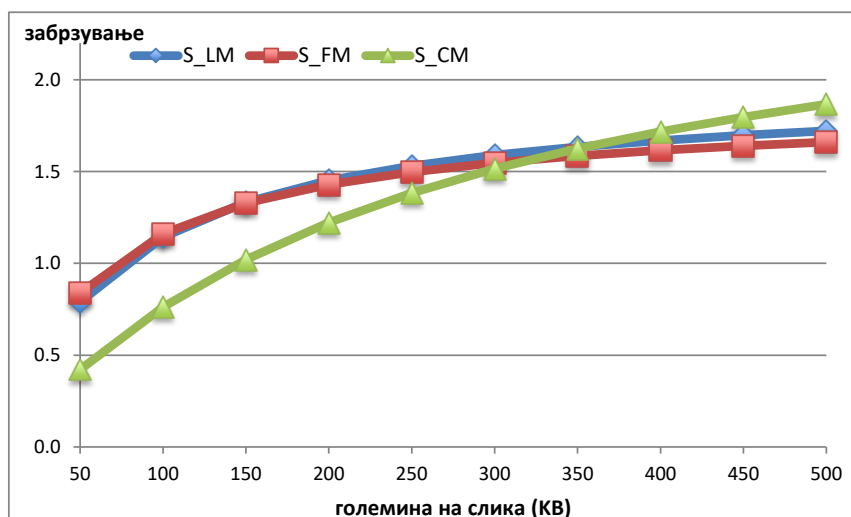
N_C	Acc_{skew}/Acc_{normal}	Acc_{blur}/Acc_{normal}
50	0,959	0,714
100	0,969	0,667
200	0,953	0,647
400	0,954	0,616
600	0,959	0,584
1000	0,959	0,562
2000	0,968	0,545
3000	0,961	0,523
4000	0,939	0,502
5000	0,932	0,486
8000	0,961	0,490
10000	0,960	0,492

6.5 Зависност од обемот на податоци за обработка

За да се определи кое од веќе дефинираните сценарија е најдобро, направени се проверки на забрзувањето помеѓу истите. Слика 6.13 и Слика 6.14 го претставуваат забрзувањето S_{LM} , S_{FM} и S_{CM} помеѓу пристапот со користење на мобилниот уред и другите сценарија (облаче, магла(3G) и облак) за мали и големи слики соодветно.



Слика 6.13: Забрзување спроти мобилното решение за мали слики

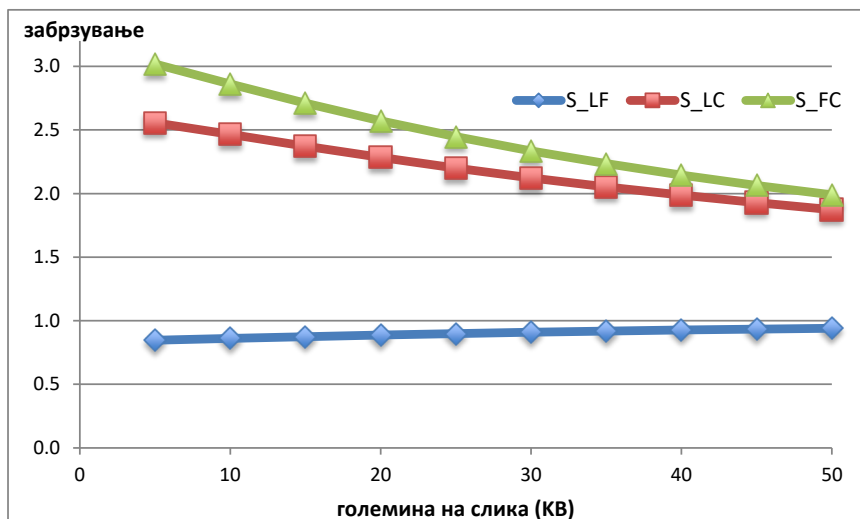


Слика 6.14: Забрзување спроти мобилното решение за големи слики

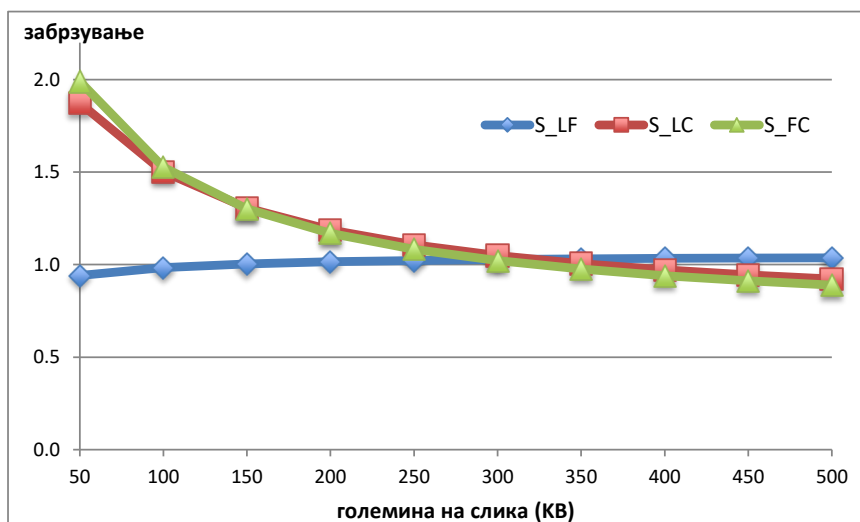
Во случај кога се користат слики со мала и средна големина мобилното решение покажува супериорни перформанси споредбено со сите анализирани пристапи. Пристапот со користење на облаче и магла добива забрзување поголемо од 1 кога големината на сликата е поголема од 75КВ. Пристапот со користење на облак е подобар од мобилното решение кога големината на сликата е поголема од 150КВ.

За да се споредат сценаријата кои го пренесуваат процесирањето на оддалечен сервер пресметани се соодветните забрзувања. Слика. 6.15 и 6.16 ги претставува забрзувањата S_{LF} , S_{FC} и S_{LC} .

Слични перформанси помеѓу пристапите со пресметување во магла (3G) и облаче се потврдени со постигнатите забрзувања. Просечната вредност изнесува 1, за слики со големина до 150КВ, пристапот со пресметување во магла (3G) е малку подобар, додека за поголеми слики подобар е пристапот со пресметување во облаче. Забрзувањето на



Слика 6.15: Забрзување помеѓу различните сценарија за мали слики



Слика 6.16: Забрзување помеѓу различните сценарија за големи слики

пристапите на пресметување во магла (3G) и облаче споредбено со пристапот со пресметувањето во облак покажува дека решението со пресметување во облак е супериорно за слики поголеми од 350KB.

6.6 Анализа на потрошувачката на батерија

Потрошувачката на батерија кај мобилните уреди е еден од главните проблеми посебно кога се користат апликации кои побаруваат повеќе ресурси од мобилниот уред. За да се провери колкава е потрошувачката на батерија кај мобилниот уред измерена е истата кај три различни мобилни уреди кога се прави процесирање на сликата и OCR процесирање и резултатите се претставени во Табела 6.4.

При пресметувањето на потрошувачката на батеријата, се користат слики со раз-

Табела 6.4: Анализа на потрошувачката на батерија

Број на OCR процесирања	Потрошувачка на батерија							
	1%	5%	10%	20%	30%	50%	70%	100%
<i>Nexus 4</i>	7	35	78	124	171	289	406	718
<i>Nexus 5</i>	10	52	109	220	316	515	703	1127
<i>Huawei P9</i>	31	149	332	651	958	1623	2235	3459

лична големина, различна комплексност, различен број на карактери и фонтови. За да бидат опфатени сите сценарија, сликани се слики од различен агол, различно позадинско светло како и заматени слики. За да се направи подетална анализа на потрошувачката на батеријата, користени се слики со големина помеѓу 5КВ и 500КВ и различна резолуција почнувајќи од 557x226 па се до 1895x787 пиксели. Исто така бројот на карактери варира помеѓу 50 и 10000 карактери. Резултатите презентирани во Табела 6.4 ги содржат просечните вредности од мерењата кои опфаќаат пет пати повторување на тестовите и притоа користење на мобилниот телефон секогаш со полна батерија.

Исто така направена е анализа кога се користат многу слики со големина од 1МВ или поголеми. Од резултатите може да се забележи дека времињата на процесирање на сликата кај сите сценарија се многу поголеми, во некои случаи дури и двојно поголеми од времињата за процесирање на слики со големина од 500МВ. Како пример ќе го земам сценариото со користење само на мобилниот уред каде најдоброто процесирачко време за слика со големина од 500МВ е 0.614 секунди, додека кај слика со големина од 1МВ времето на процесирање е 1.213 секунди.

Потрошувачката на батерија е многу поголема кога се користат слики со големина поголема од 1МВ. Споредбено со сликите со големина од 500КВ или помали, потрошувачката на енергија е за двојно поголема. Ова ја објаснува експоненцијалната зависност на потрошувачката на енергија од големината на сликата, поради зголемената обработка на барањата предизвикана од сложеноста на алгоритмот.

Од добиените резултати се гледа дека потребни се околу 7 OCR процесирања на сликата за да се потроши 1% од батеријата на мобилниот уред кога се користи Nexus 4 мобилен уред. Околу 10 процесирања се потребни кога се користи Nexus 5 мобилниот уред за процесирање, додека најдобри перформанси се постигнуваат кога се користи Huawei P9 мобилниот уред за процесирање.

6.7 Анализа на добиените резултати

Од добиените резултати може да се заклучи дека за поголеми слики потребно е повеќе време за процесирање. Исто така може да се види дека постои зависност на процесирачкото време од големината на сликата и бројот на карактери што треба да се препознаат на истата. Доколку се споредува комплексноста на сликата тогаш се гледа дека во некои случаи процесирачките времиња се скоро идентични, но во случај кога се користат слики со голем број на карактери тогаш времето на процесирање, особено кај

заматени слики, е многу поголемо. Исто така, за да нема значително зголемување на времето за процесирање, сликата не треба да биде заматена и да има добар контраст. Од анализата на графиците, може да се забележи дека процесирачкото време зависи само од големината на сликата, но не и од бројот на карактери на истата. Анализирајќи ја разликата на фонтовите, може да се заклучи дека таа незначително ги зголемува процесирачките времиња.

Според резултатите добиени кога се споредува зависноста од обемот на податоци, може да се забележи дека кога се користат слики со мала и средна големина најдобро е мобилното решение, додека пристапите со користење на облаче и магла покажуваат супериорни резултати кога големината на сликата надминува 75KB. Пристапот со користење на облак е подобар од мобилното решение кога големината на сликата е поголема од 150KB.

Од анализата на потрошувачката на батерија, може да се заклучи дека истата е многу поголема кога се користат слики со големина поголема од 1MB. Споредбено со слики со големина од 500MB и помали таа е скоро двојно поголема.

Според добиените резултати може да се заклучи дека истите целосно ги потврдуваат предходно зададените хипотези односно дека најдобар пристап е оној кога се користи облаче за процесирање. Исто така може да се види дека резултатите во потполност ги оправдуваат целите кои што беа зададени, успешна имплементација на сите пристапи и овозможување на дефинираниот модел да препознава доста сложени знаци и симболи. Добиените резултати даваат дополнителен мотив за понатамошни истражувања и подобрување на постоечките пристапи а со тоа и добивање на подобри резултати.

Глава 7

Моделирање на однесувањето на времето на процесирање

Во оваа глава се моделира зависноста на времето потребно за препознавање на карактери од бројот на карактери на сликата и големината на сликата.

Анализирајќи ги резултатите претставени во Табела 6.1, може да се забележи дека бројот на карактери и операции по карактер незначително придонесуваат во пресметувањето на времето за процесирање на сликата и препознавање на карактерите и постои зависност од големината на сликата.

Проценив дека времето на одговор T од извршувањето на алгоритмот зависи од бројот на карактери N_C и големината на сликата IS како што е прикажано во (7.1), каде f и g се функции кои треба да се утврдат експериментално.

$$T = f(N_C) + g(IS) + \epsilon \quad (7.1)$$

Во оваа равенка, ϵ е константа која ги зема вредностите во даден опсег и ја претставува зависноста од другите параметри. Параметарот ϵ ги зема вредностите изразени во ms. Флукуациите од моделираната вредност се должат на зависноста од задоцнувањата на оперативниот систем или комуникациското задоцнување. Претставениот модел се базира на експерименталните резултати и користи диференцијална деривативна техника за мерење на чувствителноста за промена на излезот во однос на влезот.

Направена е анализа со два соседни тестни случаи, еден кој вклучува број на карактери $N_C = N_1$ и големина на сликата $IS = IS_1$ што произлегува од времето на одговор $T = T_1$ и друг со N_2 , IS_2 и T_2 . Промените во приказот на времето на одговор се обележани со $\Delta T = T_1 - T_2$ додека промените во влезните параметри на тестното сценарио со промена на бројот на карактери $\Delta N_C = N_1 - N_2$ и промена на големината на сликата $\Delta IS = IS_1 - IS_2$. Едноставна анализа покажа дека ако времето за одговор е моделирано според (7.1), дериватот е еднаков на сумата од дериватите на f и g како што е прикажано во (7.2). Задачата е да се најде функција f и g или, како дериватот Δf зависи од промената на ΔN_C , и дериватот Δg од промената на ΔIS .

$$\begin{aligned}
 \Delta T &= T_1 - T_2 \\
 &= f(N_1) + g(IS_1) + \epsilon - f(N_2) - g(IS_2) - \epsilon \\
 &= \Delta f + \Delta g
 \end{aligned} \tag{7.2}$$

Дериватот го одредува наклонот на функцијата и зависноста од бројот на карактери N_C и големината на сликата IS .

7.1 Моделирање на зависноста од бројот на карактери

За да се одреди f , има две соседни тестни сценарија, големината на сликата е фиксна и $\Delta IS = 0$ води кон $\Delta g = 0$. Овде се анализира тестно сценарио со константна големина на слика IS и се пресметува дериватот од две измерени времиња T_1 и T_2 за слики со различен број карактери, соодветно обележани како N_1 и N_2 . Моделирањето е развиено според (7.3) и (7.4).

$$\Delta T = \Delta f = f(N_1) - f(N_2) \tag{7.3}$$

Претпоставувајќи дека f е линеарна функција, така што $f = \alpha N$ може да се одреди

$$\alpha = \frac{T_1 - T_2}{N_1 - N_2} = \frac{\Delta T}{\Delta N_C} \tag{7.4}$$

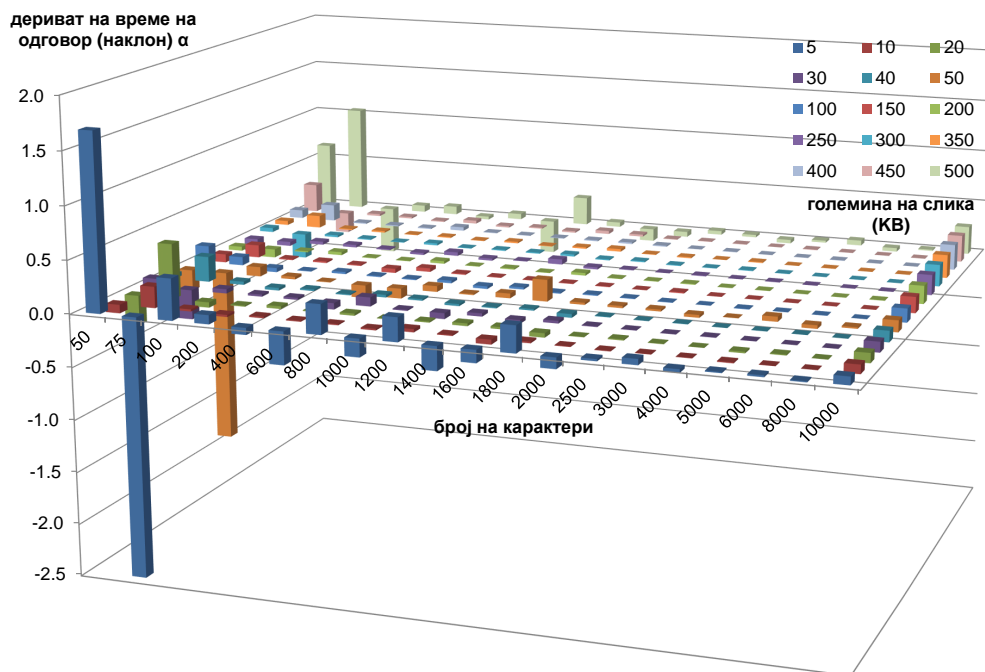
На пример, Слика 7.1 го претставува односот на наклонот на дериватот $\frac{\Delta T}{\Delta N_C}$, што е коефициент α зависен од бројот на карактери N_1 за фиксна големина на слика.

Вредностите претставени на Слика 7.1 се однесуваат на решението кога се користи само мобилниот уред за процесирање, но кај сите други решенија се добиваат слични резултати.

Севкупната анализа на однесувањето на алгоритмот врз целиот спектар на вредности покажа дека времето за одговор на имплементираните Tesseract алгоритам не зависи од бројот на карактери, т.е. дека $\alpha = 0$ и f е нулта функција. Иако резултатите покажаа мали флукуации, ова се незначителни промени, особено на границите.

7.2 Моделирање на зависноста од големината на сликата

За да се одреди g има два соседни тестни случаи така што бројот на карактери е фиксен $\Delta N_C = 0$ и дериватот е $\Delta f = 0$. Во овој случај, бројот на карактери N_C е ист и се одредува разликата помеѓу две измерени времиња на одговор T_1 и T_2 за различни големини на слики IS_1 и IS_2 .


 Слика 7.1: Определување на наклонот (α) vs N_C

Линеарно моделиран пристап сличен на моделирањето на зависноста од бројот на карактери се пресметува според (7.5) и (7.7).

$$\Delta T = \Delta g = g(IS_1) - g(IS_2) \quad (7.5)$$

Под претпоставка дека g е линеарна функција претставена преку (7.6), може да се одреди (наклонот) β со (7.7).

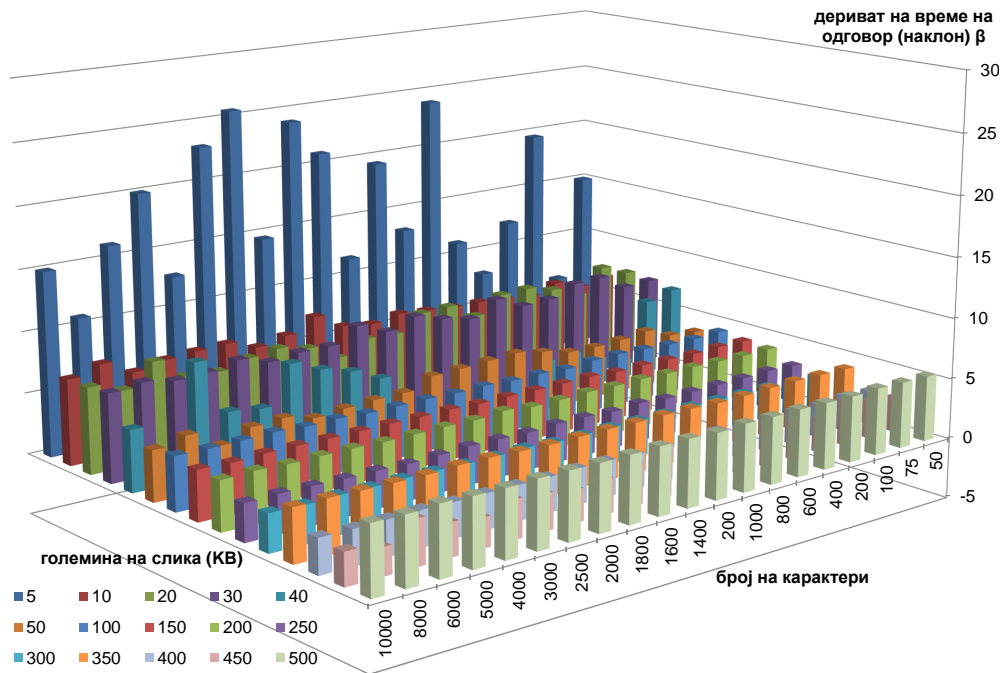
$$T = \beta \times IS + \epsilon \quad (7.6)$$

$$\beta = \frac{T_1 - T_2}{IS_1 - IS_2} = \frac{\Delta T}{\Delta IS} \quad (7.7)$$

Слика 7.2 го претставува наклонот на g , кој е параметарот β зависен од големината на сликата за фиксен број на N_1 карактери

Може да се забележи дека има големи флукуации во опсегот до 50KB, кој е поради компресираниот алгоритам и зависноста од бројот на карактери во компресираната слика. Всушност, ова покажува експоненцијално намалување на параметарот β .

Однесувањето на зависноста на големини на слики поголеми од 50KB покажува прилично стабилно однесување каде што приближувањето на β може да се одреди на просечна вредност од $\beta_M = 4.28$ кај мобилната самостојна апликација, $\beta_L = 1.55$ за решението кое користи облаче за процесирање, $\beta_F = 2.06$ за пресметувањето во магла и исто така кај пресметувањето во облак. Овие вредности приближно одговараат на слики со големина од 100KB. За помали слики, вредноста на β благо се зголемува, а за



Слика 7.2: Дериват на времето на одговор (наклон) β vs N_C и IS

поголеми слики благо се намалува. Флукуациите се исто така присутни тука поради комуникациското задоцнување кое е зависно од тековното оптоварување на мрежата.

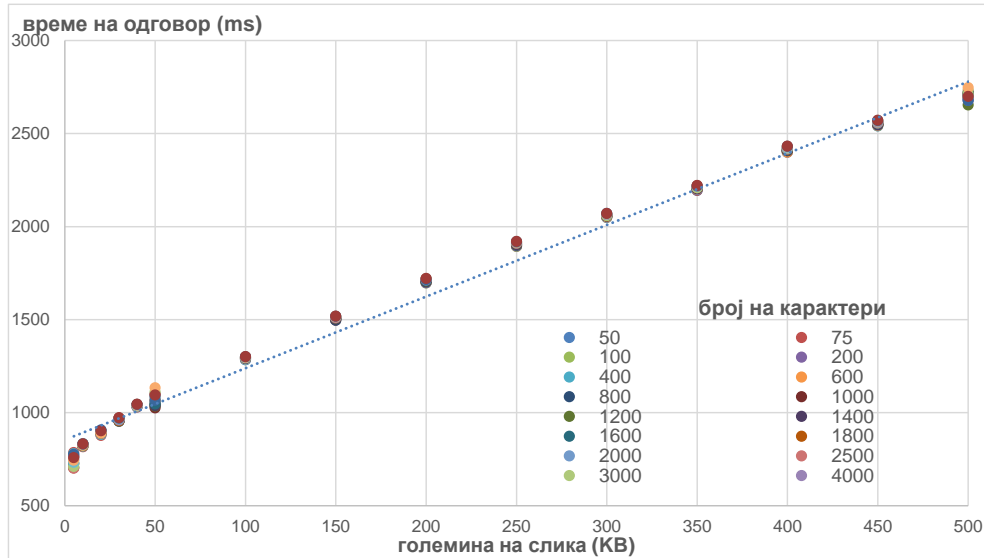
Вредностите на параметарот β се изразени во ms/KB . Вредностите на ϵ може лесно да се пресметаат со замена на вредноста на β во равенката со времето на одговор T и големината на сликата IS .

Илустрација на моделираната зависност на функцијата (7.6) е прикажана на Слика 7.3

Големите флукуации подразбираат дека g не е линеарна функција зависна од IS . Според тоа (7.6) не е валидна, што значи дека дериватот на функцијата не го моделира побараното време на одговор. Стандардната деривација на пресметаните времиња на одговор за преддефинираните големини на слики е 60 додека вкупната сума на разликата е -50, што покажува дека моделираната функција само го приближува времето за одговор во мал интервал.

Бидејќи алгоритамската комплексност е $O(\log N)$ поради зголемената искористеност на трите структури во пресметките, претставени се предлог функции за моделирање со следните функции на линеарни и логаритамски функции:

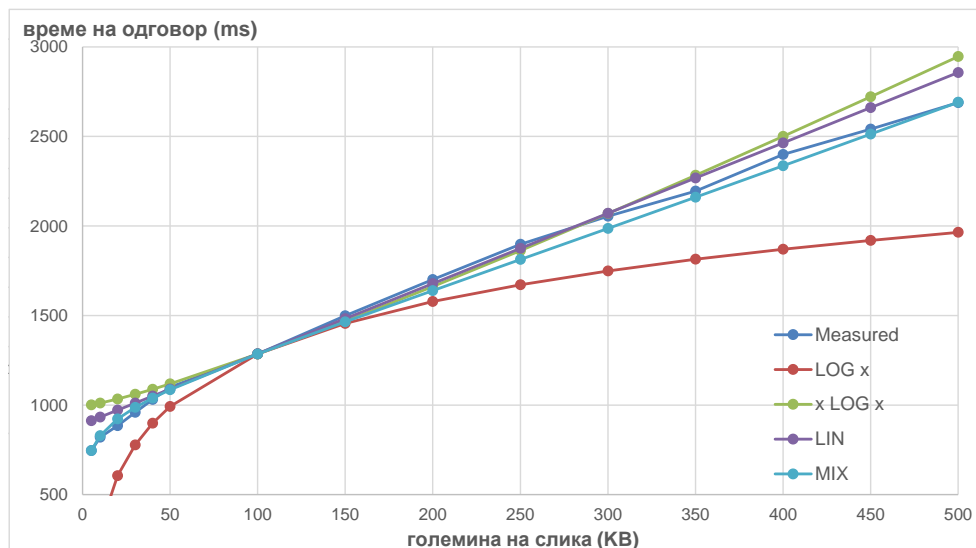
- **LIN** е линеарна функција на времето на одговор $T_{LIN} = \beta \times IS + \epsilon_1$
- **LOG x** е логаритамска функција на времето на одговор $T_{LOG} = \gamma \times \log(IS) + \epsilon_2$
- **x LOG x** е логаритамска функција на времето на одговор $T_{xLOG} = \delta \times IS \times \log(IS) + \epsilon_3$
- **MIX** е мешавина од две логаритамски функции на времето на одговор $T = \gamma_2 T_{LOG} + \delta_2 T_{xLOG}$.



Слика 7.3: Тренд на измерените вредности и моделираната функција

Се забележува дека параметрите β , γ , δ и ϵ се избрани за да одговараат на функцијата на времето на одговор за преддефинирана вредност на големината на слика од 100KB. Вредностите на γ_2 и δ_2 се одбрани за да одговараат на функцијата соодветно со вредности од 0.26 и 0.74. Конечните пресметани коефициенти водат до моделирачка функција претставена во (7.8).

$$T = 1.069 \times IS \times \log(IS) + 252.7 \times \log(IS) + 565.7 \quad (7.8)$$



Слика 7.4: Приближување на моделираните функции на зависност на времето на одговор од големината на сликата

Вредностите на добиените моделирачки функции и пресметаните вредности се претставени на Слика 7.4. Добиените резултати за стандардното отстапување се до 30

за опсег на вредности помеѓу 750ms и 2700ms. Максималната грешка е еднаква на максимум 3% од пресметаната вредност, што дава добро приближување. Флукуациите во добиените времиња на одговор и моделираната функција може да се повторат поради моменталниот статус на пресметувачката околина. Оперативниот систем или другите позадински процеси исто така може да направат разлики во очекуваните перформанси. Овој моделирачки процес се однесува само на просечните перформанси кога не се извршуваат други процеси паралелно.

Параметрите во (7.8) се зависни од процесорот, брзината и меморијата на компјутерот. Исто така, вредностите се зависни од тестната околина, па кај пресметувањето користејќи облак, облаче или магла за процесирање може да се очекуваат различни коефициенти.

Однесувањето на перформансите е претставено во (7.9) како генерички модел за однесување на перформансите. Параметрите λ и μ може да се утврдат експериментално во зависност од пресметувачката околина

$$T = \lambda(IS + 236.4)\log(IS) + \mu \quad (7.9)$$

Комплексноста на алгоритмот е пресметана со $O(n \log(n)) + O(\log(n))$.

7.3 Анализа на најдоброто решение

Теоретските резултати покажаа дека најдобро решение каде се користи оддалечен сервер за процесирање е пристапот со користење на магла со 4G мрежно поврзување. Експерименталниот дел обработен во докторската дисертација вклучува користење само на 3G мрежно поврзување кај ова сценарио. Овој пристап овозможува побрза комуникација со најниските теоретски и експериментално пресметани задоцнувања, што го прави ова решение најдобро при користење на оддалечен сервер.

Моментално, мобилните оператори сеуште се незаинтересирани од користење на ваков пристап за комерцијални решенија, па поради тоа, решението со користење на облаче за процесирање е најприфатливо. На пример, модерните шопинг центри, дефинитивно би го понудиле ова решение како дополнителен сервис на своите корисници кој со својата флексибилност и перформанси ги зачувува ресурсите на мобилни телефони на корисниците што е можно повеќе.

Исто така оваа докторска дисертација го потврди заклучокот дека перформансите на OCR процесирањето изразени преку вкупното време на одговор, најмногу зависат од големината на сликата. Од резултатите во докторската дисертација може да се заклучи дека, за препознавање на карактери при користење на мал фонт и мали слики, најдобро решение, анализирајќи ги перформансите само на времето за одговор е користењето само на мобилниот уред, притоа, трошејќи ги ресурсите на мобилниот уред и зголемувајќи ја потрошувачката на батерија. Кога сликите се комплексни или се поголеми, секогаш е најдобро процесирањето да се извршува на најблиското облаче или на оддалечен облак. Овој пристап ја штеди потрошувачката на батерија.

Предложената стратегија за одбирање на оптимално решение е базирана на експерименталните резултати добиени користејќи го мобилниот уред, облачето во близина на мобилниот уред, и облакот сместен во Орегон (USA). Направени се три сценарија за да се провери кое од предложените решенија е најдобро.

Иако во докторската дисертација се внесени само три типови на паметни телефони, направени се експерименти и со неколку други паметни телефони. Всушност, тестните уреди припаѓаат на еден од трите дефинирани класи на паметни телефони, ниска класа, каде телефонот е со вредност под 100 долари, средна класа и висока класа. Huawei P9 припаѓа на телефоните од висока и моќна класа и прикажа најдобри резултати. Nexus 5 и Nexus 4 соодветно припаѓаат на телефони од средна и ниска класа, не само поради нивната вредност, туку поради процесирачката моќ и капацитетот на батеријата.

7.4 Споредба со други пристапи

Познавајќи го фактот дека постојат многу други различни пристапи кои се занимаваат со истата тематика како оваа претставена во докторската дисертација, направена е споредба за да се провери кое од сите тие решенија вклучувајќи го и моето дава најдобри резултати. Најблизок пристап до мојот е претставен во [45] каде Mei и останатите ја објаснуваат предноста од префрлување на апликацијата на виртуализирана околина на облакот. Тие ги пресметуваат перформансите на протокот на податоци и ефикасноста од споделувањето на ресурсите, вклучувајќи го и влијанието на неактивните инстанци на апликацијата кои работат истовремено на истиот физички сервер. Исто така тие истражуваат како различно планираните CPU ресурси влијаат врз перформансите на виртуелниот сервер. Предноста на моето решение над нивното е тоа што тие ги изведуваат ваквите експерименти користејќи го само облакот како платформа додека во мојата дисертација се испитуваат различни архитектурни решенија и се одредува најдоброто, што дава поголема флексибилност и повеќе опции за извршување на апликацијата.

Повеќето од решенијата се базирани на процесирање на сликата и препознавање на карактерите на истата, главната разлика е каде се прави процесирањето. Кога процесирањето се прави на мобилниот уред нема голема разлика во времињата за одговор. Но кога треба сликата да се префрли на најблиското облаче или оддаелечен облак тогаш постои разлика во мрежните задоцнувања и протокот на податоци. Garcia и останатите [29] објаснуваат како различните вредности за задоцнувањето може да дадат значително различни резултати. Исто така нивното истражување го пресметува вијанието на задоцнувањата во податочниот сообраќај, претставувајќи ја важноста од пресметување на задоцнувањето при оптовареност на мрежата.

Caushaj и останатите во [30] го споредуваат протокот на податоци и задоцнувањето кај 3G и 4G мрежите и добиваат незначително подобри резултати. Ова може да се должи на не толку добрата конфигурација на мобилната мрежа, различната фреквенција на мобилната мрежа како и инфраструктурата во двата случаи. Исто така како што е

наведено во [46] користењето на Wi-Fi дава супериорни резултати споредбено со користењето на LTE ако се гледаат параметрите за проток на податоци и задоцнување, што е исто така докажано и во мојата работа.

Во поголем број на случаи, протокот на податоци и задоцнувањето зависат од бројот на станици на бежичната мрежа кога се користи класичниот метод за пристап до мрежата. Доколку мрежата се состои од мал број на станици и мала должина на рамките, класичниот метод за пристап постигнува далеку најдобри резултати за проток на податоците.

Parageorgiou и останатите во [47] претставуваат пристап кој всушност е способен да ги избегне реалните задоцнувања. Тие се осврнуваат на важноста од топологијата на надворешните интеракции, како на пример интеракции со податочната база, корисниците, итн. Нивното решение може целосно да ги елиминира задоцнувањата и да го зголеми пропусниот опсег.

Satyanarayanan во [48] објаснува како да се избегне задоцнувањето при процесирање на податоци кога се користи облаче како сервер за процесирање и како да се намали пропусниот опсег во облакот. Тој објаснува како префрлувањето на процесирањето на облачето или облакот може да биде од голема корист за општеството. Како пример зема модерен авион кој може да генерира речиси половина терабајт податоци од вградени сензори за време на летот. Анализата во реално време на овие податоци на облачето во авионот може да даде навремени насоки за превентивно одржување, заштеда на гориво и други предности.

Предноста од доцнењето и пропусниот опсег кај облачето се особено важни кај автомобилите, посебно кај пристапите на возило до возило, каде придонесува за контрола во реално време и избегнување на незгоди. Во блиска иднина, поврзаноста со облакот од автомобил во движење ќе биде преку 3G или 4G/LTE мрежно поврзување. Важно прашање е дали облачето ќе биде сместено во самиот автомобил или пак ќе биде дел од инфраструктурата на телекомуникацискиот оператор (на пример едно облаче да биде конектирано со повеќе базни станици во околината).

Simoes и останатите презентираат скалабилен Интернет систем за континуирано собирање на видеа од луѓе од уреди како што се Google Glass [49]. Нивната хибридна архитектура на облакот, GigaSight, е ефикасна мрежа за испорака на содржини во обратна насока. Таа постигнува скалабилност преку децентрализација на инфраструктурата за собирање со користење на облаче сместено на виртуелни машини. Во нивниот прототип, намалувањето на резолуцијата, не ја намалува точноста при детектирање на објекти, но драматично ја намалува точноста на алгоритмот за препознавањето на лица.

Mishra и останатите, презентираат во [36] OCR мобилна апликација која користи позадински сервери за процесирање. Предноста на решението претставено во докторската дисертација со решението претставено во [36] е тоа дека дозволува избор каде ќе се прави процесирањето, на мобилниот уред или пак на серверот.

Jagarweh и останатите, во [50] претставуваат решение за мобилно пресметување на облакот користејќи инфраструктура базирана на облаче се со цел намалување на пот-

рошувачката на енергија и мрежното задоцнување и задоволување на договорот за ниво на услуга (SLA). Тие сметаат дека станува збор за развој на големи корпоративни облачиња и поради тоа предлагаат модел за решавање на проблемот со оптимизација на потрошувачката на енергија за такви системи. Овој пристап може да се користи како напредно истражување на решението претставено во докторската дисертација за решавање на проблемот со оптимизација на потрошувачката на енергија.

Chen и останатите [51] претставуваат решение кое е слично на тоа претставено во докторската дисертација. Тие имаат развиено рамка за поддршка на пресметување кое го зема во предвид контекстот на апликации кои се извршуваат во реално време. Користат четири најблиски облачиња и еден јавен сервис на облакот, кои може да се користат за одложено пресметување во различни мрежи. Три видови на апликации се користат во нивниот експеримент, и тоа основна апликација, апликација со традиционално одложување на пресметувањето и апликација која при пресметување го зема во предвид контекстот. Основната апликација се извршува целосно на телефонот, апликацијата со традиционално одложување на пресметувањето го користи само јавниот сервис на облакот за пресметување, додека апликацијата која при пресметување го зема во предвид контекстот може да користи било кој од погоре споменатите ресурси за пресметување. Резултатите покажуваат дека нивниот пристап може значително да помогне во намалување на времето потребно за извршување на апликацијата и заштеда на потрошувачката на батерија што е и заклучок од докторската дисертација.

XiaoFan и останатите [46] ја разгледуваат брзината на јазлите, просечното растојание на чекорите и бројот на јазли вклучени во мрежата со цел да ги одредат перформансите на двете метрики, пропусниот опсег и задоцнувањето во 3G/4G мрежата. Споредувајќи ги перформансите на LTE и Wi-Fi мрежите, може да се забележи дека Wi-Fi постигнува повисока стапка на пропусност и помало задоцнување споредбено со LTE.

Исто така, направена е споредба на задоцнувањето и стапките на пренос споредувајќи ги WAN и LAN мрежите. LAN вообичаено нуди висока стапка на пренос на податоци кај мали растојанија помеѓу крајните точки, додека WAN нуди пониска стапка на пренос на податоците. Растојанијата обично влијаат на стапката за пренос на податоците дури и ако тие се еднакви на LAN (10/100/1000 Mbps). Доцнењето е многу поголемо кај WAN поради далечината и бројот на уреди низ кои податоците треба да поминат.

Протоколот на податоци и задоцнувањето може да играат главна улога во процесирањето на сликите. За мали слики протоколот не прави голема разлика, особено кога се користат слики со големина од 5KB и 50KB, но задоцнувањето влијае на перформансите. Во случај кога се користат големи слики (поголеми од 500KB), тогаш протоколот исто така влијае на целосното комуникациско задоцнување а со самото тоа и на целото процесирање.

Глава 8

Моделирање на перформанси за различни архитектурни решенија

Моделирањето на перформансите за различни архитектурни решенија се претставени преку процес кој ги опфаќа различните сценарија кои даваат решение на мотивациониот проблем.

8.1 Анализа на перформансите

Во овој дел се дефинирани процесирачките времиња на секоја активност поодделно, Табела 8.1

Табела 8.1: Времиња на одговор за секоја дефинирана активност

активност	идентификација
<i>Процесирање на слика</i>	T_I
<i>OSR процесирање</i>	T_O
<i>Префрлување на сликата</i>	T_T
<i>Добивање на одговор</i>	T_R

За да се идентификуваат сценаријата и локацијата на секое процесирање, доделени се "M" за мобилен, "C" за облак, "F" за магла и "L" за облаче како втора буква во идентификацијата. На пример, T_{IM} значи пресметување на времето на одговор за процесирање на слика на мобилен уред. Времињата на одговор за секое сценарио идентификувано со "M", "C", "F", "L" се дефинирани во равенката (8.1), претпоставувајќи дека активностите за сликање на сликата и презентирање на резултатот назад на мобилниот уред се исти за сите сценарија.

$$\begin{aligned} T_M &= T_{IM} + T_{OM} \\ T_L &= T_{TL} + T_{RL} + T_{IL} + T_{OL} \\ T_F &= T_{TF} + T_{RF} + T_{IF} + T_{OF} \end{aligned} \tag{8.1}$$

$$T_C = T_{TC} + T_{RC} + T_{IC} + T_{OC}$$

8.2 Задоцнување во комуникацијата

Времето на префрлување на сликата кај сите сценарија, е еднакво на сумата од иницијалното комуникациско задоцнување дефинирано од мрежата и времето потребно за префрлување на потребните податоци дефинирани во (8.2).

$$T_T = Network_Delay + Transfer_time \quad (8.2)$$

Времето потребно за префрлување на податоците се пресметува како однос помеѓу големината на сликата IS (во бајти) и брзината на мрежата (во бајти во секунда), како што е дефинирано во (8.3).

$$Transfer_time = \frac{IS}{Network_speed} \quad (8.3)$$

За поедноставување, се користи D за *мрежно задоцнување*, изразувајќи го мрежното задоцнување при трансфер на податоци. Според тоа, вредностите за вкупното време за трансфер на сликата кај секое од сценаријата се дефинирани во (8.4), претпоставувајќи дека мрежното задоцнување е дефинирано од Wi-Fi, 3G/4G и WAN/LAN соодветно за облаче, магла и облак процесирачките сценарија.

$$\begin{aligned} T_{TL} &= D_{WiFi} + \frac{IS}{WiFi_speed} \\ T_{TF} &= D_{3G/4G} + \frac{IS}{3G/4G_speed} \\ T_{TC} &= D_{WAN} + \frac{IS}{WAN_speed} \end{aligned} \quad (8.4)$$

Претпоставувајќи дека времето на одговор е многу мало множество на податоци споредено со големината на сликата и претпоставувајќи дека нема потреба да се додава било какво време за префрлување при процесот на враќање на резултатот назад на мобилниот уред, времето на одговор ќе биде исто како и времето на мрежното задоцнување. (8.5) ги дефинира времињата на враќање на резултатот назад на мобилниот уред. Потоа, се анализираат и се споредуваат нивните вредности преку метриците за задоцнување и проток на податоци.

$$\begin{aligned} T_{RL} &= D_{WiFi} \\ T_{RF} &= D_{3G/4G} \\ T_{RC} &= D_{WAN} \end{aligned} \quad (8.5)$$

Целосното задоцнување во комуникацијата за префрлување на сликата и добивање на резултатот назад на мобилниот уред е еднакво на соодветниот збир како што е прикажано во (8.6).

$$\begin{aligned}
 T_{comm}(L) &= T_{TL} + T_{RL} = 2 * D_{WiFi} + \frac{IS}{V_{WiFi}} \\
 T_{comm}(F) &= T_{TF} + T_{RF} = 2 * D_{3G/4G} + \frac{IS}{V_{3G/4G}} \\
 T_{comm}(C) &= T_{TC} + T_{RC} = 2 * D_{WAN} + \frac{IS}{V_{WAN}}
 \end{aligned} \tag{8.6}$$

За да се добие попрецизно и точно време на одговор, мора да се пронајде точното задоцнување и проток на мрежата преку која се прави префрлувањето на сликата. Во Табела 8.2 се претставени крајните вредности на задоцнувањето и протокот на податоци на анализираните мрежи [30], [52].

Табела 8.2: Типичен проток на податоци V и задоцнување D за Wi-Fi/3G/4G/WAN мрежи

Комуникација	V (Mbit/sec)	Min D (sec)	Max D (sec)
<i>Wi-Fi (802.11a)</i>	54	0.1225	0.2805
<i>Wi-Fi (802.11b)</i>	11	0.2785	0.5280
<i>Wi-Fi (802.11g)</i>	54	0.5230	0.9430
<i>3G (HSPA+)</i>	168	0.1000	0.5000
<i>4G (LTE)</i>	326.4	0.0600	0.1000
<i>WAN</i>	>100	0.3140	0.8000

Следната анализа претпоставува дека протокот и задоцнувањето во моите сценарија користејќи Wi-Fi, 3G или 4G мрежа ги достигнува максималните вредности дефинирани од самата мрежа. Поради тоа, моделирачкиот процес ги анализира најдобрите и најлошите сценарија. Комуникациското задоцнување како сума на префрлувањето на сликата и враќање на резултатот назад на мобилниот уред пресметани како $T_T + T_R$ се претставени во Табела 8.3. Таму се анализирани три основни вредности за големини на слики: 5KB за *мали слики*, 50KB за *средни слики* и 500KB за *големи слики*.

Табела 8.3: Најдобри и најлоши случаи за мрежно задоцнување

	Мали 5KB	Средни 50KB	Големи 500KB
<i>Облаче</i>	0.246 - 0.571	0.252 - 0.577	0.319 - 0.644
<i>Магла (3G)</i>	0.200 - 1.000	0.202 - 1.002	0.224 - 1.024
<i>Магла (4G)</i>	0.120 - 0.200	0.121 - 0.201	0.132 - 0.212
<i>Облак</i>	0.628 - 1.600	0.632 - 1.604	0.668 - 1.640

Вредностите во Табела 8.3 се добиени со користење на вредностите од Табела 8.2 и равенката 8.6. На пример, најдобриот случај користејќи облаче за процесирање се

добива кога се користи проток на податоци $V_{WiFi} = 54/8 = 6750KB/sec$ и минимално задоцнување $D_{WiFi} = 0.1225$ и со тоа се пресметува целосното комуникациско задоцнување за големина на слика од 5KB со $T_{TL} + T_{RL} = 2 * D_{WiFi} + (IS/V_{WiFi}) = 2 * 0.1225 + (5/6750) = 0.2457$.

Протоколот на податоци и задоцнувањето може да играат главна улога при процесирање на сликите. За мали слики протокот не прави голема разлика, особено кога се користат слики со големина од 5KB и 50KB. Но кога се користат слики со големини од 500KB и поголеми, тогаш протокот и тоа како влијае врз комуникациското задоцнување, а со тоа и врз целосното процесирање на сликата.

8.3 Анализа на зависноста од бројот на процесирани карактери

Времето на одговор го одразува однесувањето на алгоритмот за процесирање на сликата и препознавањето на карактерите. Времето на одговор е тесно поврзано со големината на сликата, бројот на карактери за процесирање и директно зависи од процесирачката моќ на серверите на кои се извршува процесирањето.

Табелите 8.4 и 8.5 ја претставуваат процесирачката моќ во MIPS (милиони инструкции во секунда) на секој од мобилните уреди и сервери кои се користат за процесирање. Вредностите се земени од PassMark веб страната која ги содржи процесирачките вредности на повеќе од 800.000 процесори [53]. Резултатите се ажурираат дневно па нивната вредност варира од ден на ден. Селектирани се три класи на мобилни уреди и тоа нискобуџетни, уреди од средна класа и уреди од висока класа со високи перформанси. Во случајот со серверите избрани се сервери од средна класа достапни за експериментите извршени на крајот од 2016 година.

Табела 8.4: Процесирачка моќ на мобилните уреди користени во експериментите

Класа	Мобилен уред	MIPS
ниско буџетна класа	<i>Nexus 4</i>	1327
средна класа	<i>Nexus 5</i>	2266
висока класа	<i>Huawei P9</i>	4856

Табела 8.5: Процесирачка моќ на серверите од средна класа користени во експериментите

Сервери	MIPS
<i>Облаче</i>	5680
<i>Мајла</i>	5680
<i>Облак</i>	16520

Процесирањето на сликата и препознавањето на карактерите во експериментите користи еден ист алгоритам во сите дефинирани сценарија. Анализата на алгоритмот

покажува дека бројот на операции зависи од големината на сликата IS и бројот на карактери N_C , а со самото тоа и процесирањето на сликата и препознавањето на карактерите.

Времињата за процесирање на сликата и препознавање на карактерите зависат од:

- процесирачката моќ на серверот V изразена во број на операции во секунда,
- бројот на карактери N_C на сликата,
- операции по карактер O_C за препознавањето на карактерите,
- бројот на пиксели N_P на сликата, и
- операции по пиксел O_P за процесирање на сликата.

Треба да се забележи дека големината на сликата IS изразена во бајти е директно зависна од бројот на пиксели N_P . Во случај кога сликата е компресирана од даден формат, jpeg, png или некој друг сличен на нив, тогаш сликата мора да биде репроцесирана и вратена во првичниот формат.

Вкупното процесирачко време е збир од времето за процесирање на сликата и времето за препознавање на карактерите како што е прикажано во (8.7). Треба да се забележи дека вкупното процесирачко време вклучува и време на системот T_{system} за означување на зависноста од системот.

$$T_{proc} = T_I + T_O = \frac{(N_C * O_C)}{V} + \frac{(N_P * O_P)}{V} + T_{system} \quad (8.7)$$

Во [54] детално е анализирано однесувањето на перформансите на OCR решението и зависноста од големината, комплексноста на сликата и бројот на карактери на истата. Според [54] процесирачкото време не зависи од бројот на карактери N_C , туку зависи од големината на сликата за имплементираните алгоритми со комплексност $O(IS \log(IS))$. Тоа значи дека во равенката O_C е многу мала вредност и $N_C * O_C \ll N_P * O_P$, па истата може да се занемари во понатамошните пресметки. Покрај тоа, N_P зависи од големината на сликата и од алгоритам за компресирање на истата (на пример, jpeg, png или други). Според оваа анализа се користи (8.8), каде O_S е коефициентот кој го дефинира бројот на операции по големина на слика, во зависност од алгоритмот за компресија.

$$T_{proc} = T_I + T_O = \frac{IS * \log(IS) * O_S}{V} + T_{system} \quad (8.8)$$

Процесот на моделирање го анализира најлошото и најдоброто сценарио. Времињата за одговор при процесирање на сликата и препознавање на карактерите се пресметуваат преку (8.7) и се претставени во Табела 8.6 анализирајќи три големини на слики: 5KB за мали, 50KB за средни и 500KB за големи слики. Бројот на карактери кои се процесираат се 200, 1600 и 10000 соодветно за мали, средни и големи слики.

Табела 8.6: Најдобро и најлошо сценарио за процесирање на слика и препознавање на карактери (во секунди)

	Мали 5КВ	Средни 50КВ	Големи 500КВ
<i>LG Nexus 4</i>	0.072 - 0.287	0.540 - 2.159	2.248 - 8.991
<i>LG Nexus 5</i>	0.042 - 0.168	0.316 - 1.265	1.316 - 5.265
<i>Huawei P9</i>	0.020 - 0.078	0.148 - 0.590	0.614 - 2.457
<i>Облаче</i>	0.017 - 0.067	0.126 - 0.504	0.525 - 2.101
<i>Магла (3G)</i>	0.017 - 0.067	0.126 - 0.504	0.525 - 2.101
<i>Магла (4G)</i>	0.017 - 0.067	0.126 - 0.504	0.525 - 2.101
<i>Облак</i>	0.006 - 0.023	0.043 - 0.173	0.181 - 0.722

8.4 Вкупно време на процесирање

Вкупното време на процесирање се добива како збир од комуникациското задоцнување (8.2) и времето на одговор (8.8) за процесирање на сликата и препознавање на карактери. Во Табела 8.7 сумирани се вредностите за најдобро и најлошо време на одговор од Табела 8.3 и 8.6.

Табела 8.7: Најдобри и најлоши пресметувачки резултати (во секунди)

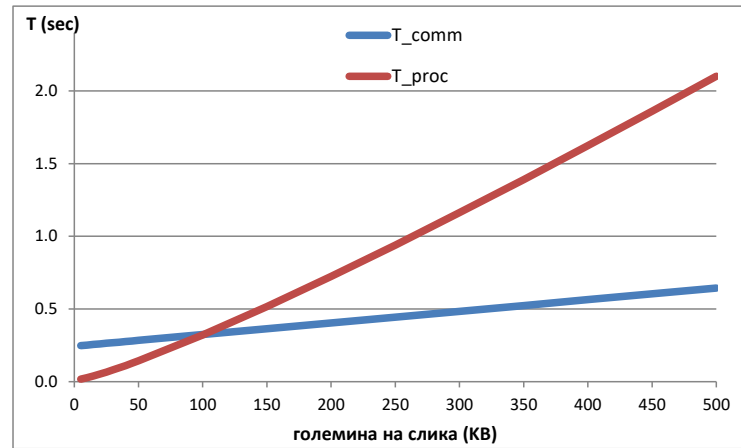
	Мали 5КВ	Средни 50КВ	Големи 500КВ
<i>Мобилен уред</i>	0.020 - 0.287	0.148 - 2.159	0.614 - 8.991
<i>Облаче</i>	0.263 - 0.638	0.378 - 1.081	0.844 - 2.745
<i>Магла (3G)</i>	0.217 - 1.067	0.328 - 1.506	0.749 - 3.125
<i>Магла (4G)</i>	0.137 - 0.267	0.247 - 0.705	0.657 - 2.313
<i>Облак</i>	0.634 - 1.623	0.675 - 1.777	0.849 - 2.362

Предходната моделирачка анализа ја поедноставува равенката (8.1) преку пресметување на вкупното време за одговор користејќи ја (8.9).

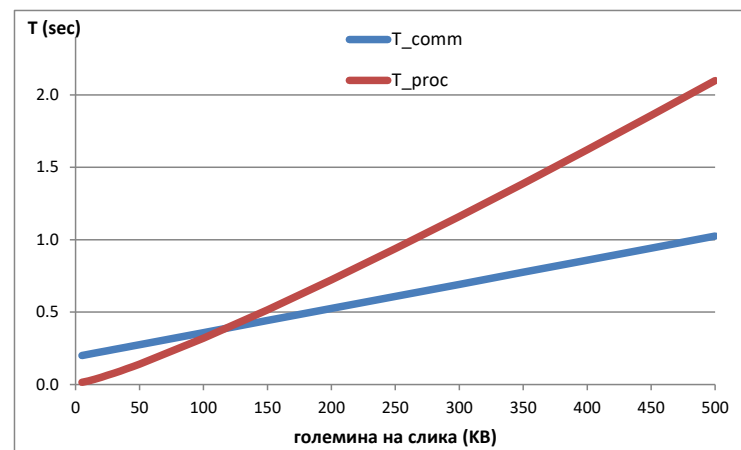
$$T = T_{comm} + T_{proc} \quad (8.9)$$

Ваквото моделирање на перформансите е претставено на Слика 8.1, Слика 8.2, Слика 8.3 и Слика 8.4 соодветно за облаче, магла (3G), магла (4G) и облак сценаријата.

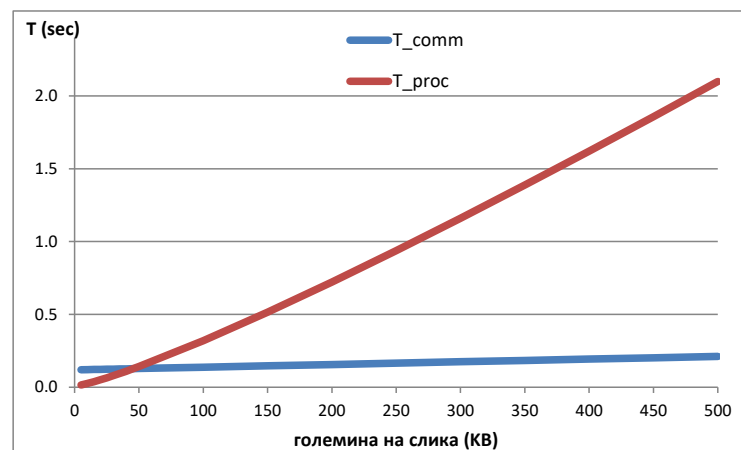
Може да се забележи логаритамското комплексно однесување на процесирачкото време и линеарната комплексност на комуникациското задоцнување. За мали слики комуникациското време е многу поголемо од процесирачкото време. Во одредена точка процесирачкото време кај пристапите со користење на облаче и магла се изедначува со комуникациското време. Кај пристапите кои користат 3G и Wi-Fi ова се случува помеѓу средни и големи слики со вредности помеѓу 100КВ и 150КВ. Треба да се забележи дека кај сценариото со користење на магла (4G), поради малото комуникациско задоцнување и големиот проток ова се случува кај слики со средна големина од 45КВ.



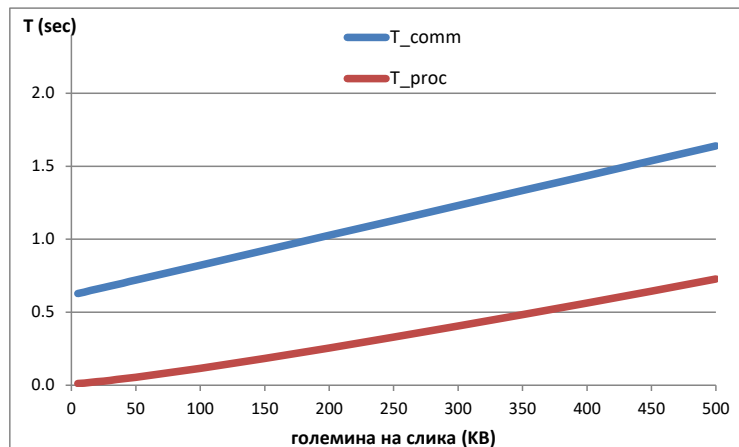
Слика 8.1: Модел на време за одговор, вклучувајќи време на комуникација и процесирање за сценарио L - облаче



Слика 8.2: Модел на време за одговор, вклучувајќи време на комуникација и процесирање за сценарио магла (3G)



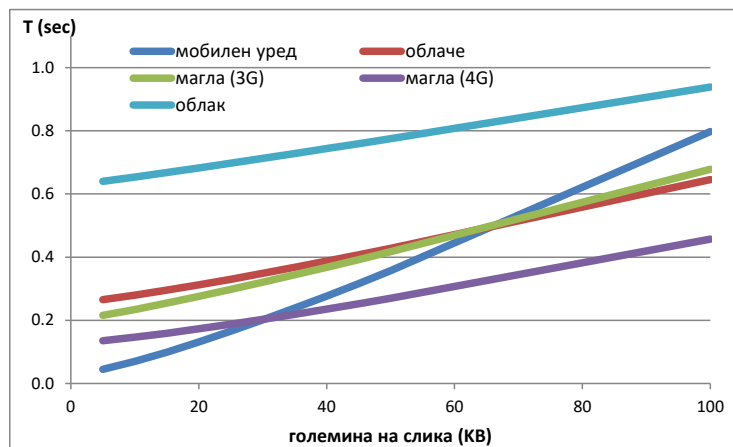
Слика 8.3: Модел на време за одговор, вклучувајќи време на комуникација и процесирање за сценарио магла (4G)



Слика 8.4: Модел на време за одговор, вклучувајќи време на комуникација и процесирање за сценарио С - облак

Уште еден интересен феномен е дека сценариото кога се користи облак за процесирање, WAN задоцнувањето е секогаш многу поголемо од процесирачката брзина така што доминира во перформансите на сценариото.

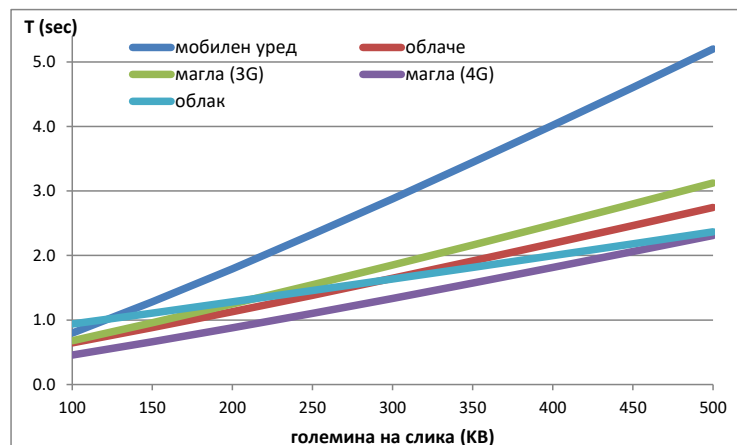
Слика 8.5 ги споредува времињата на одговор за сите анализирани пристапи за мали слики, додека Слика 8.6 ги опфаќа големите слики.



Слика 8.5: Време за процесирање на различните сценарија за мали слики

Заклучоците од теоретската анализа се следни:

- Најдобро решение за мали слики со големина до 30KB е решението со користење само на мобилниот уред. Ова е поради комуникациското задоцнување потребно за другите сценарија.
- Најдобро решение за средни и големи слики со големина поголема од 30KB е пресметувањето во магла и користење на 4G поради малото комуникациско задоцнување споредено со другите сценарија.
- Најдобро решение за многу големи слики со големина поголема од 500KB е сценариото со користење на облак за процесирање.

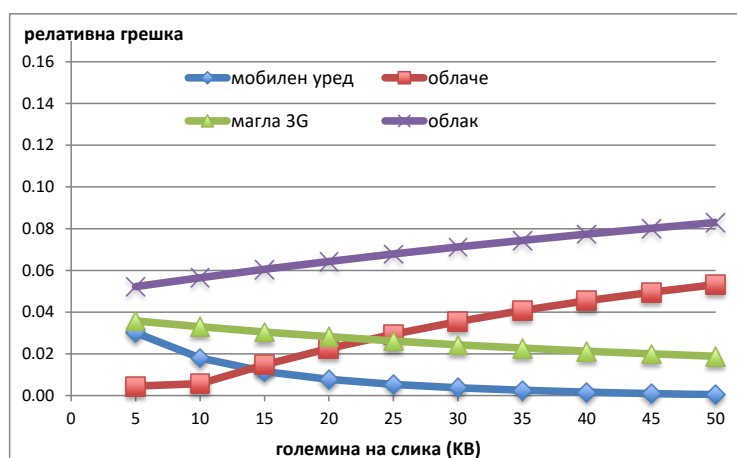


Слика 8.6: Време за процесирање на различните сценарија за големи слики

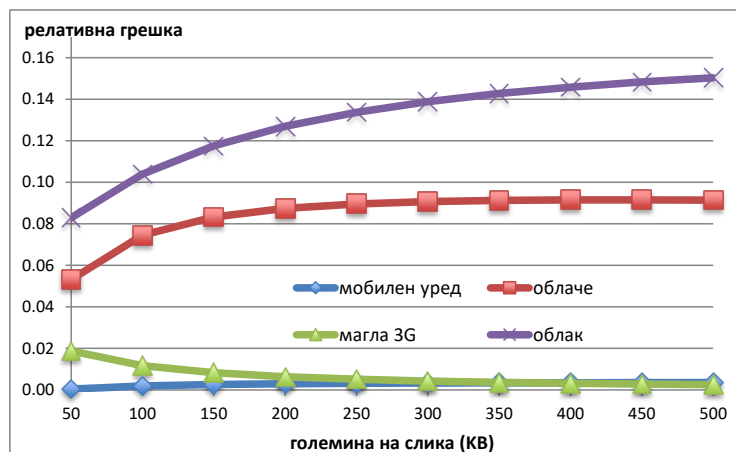
- Сценариото со користење само на мобилниот уред е лошо за слики со големина поголема од 50KB и станува многу полошо кога се користат слики со големина поголема од 120KB.
- Сценаријата со користење на облаче и магла (3G) се однесуваат скоро идентично, сценариото со користење на облаче е подобро од сценариото со користење на мобилниот уред и магла (3G) за големини на слики поголеми од 70KB.

8.5 Отстапувања помеѓу моделот и експерименталните резултати

Релативната грешка помеѓу добиените резултати од теоретскиот пристап и експерименталните резултати се претставени на Слика 8.7 и 8.8 соодветно за мали и големи слики.



Слика 8.7: Релативна грешка помеѓу различните сценарија за мали слики



Слика 8.8: Релативна грешка помеѓу различните сценарија за големи слики

Мобилното сценарио е скоро идеален модел за сите анализирани слики и најголемата релативна грешка е за многу мали слики иако е помала од 3%.

Моделот за сценариото каде се користи магла (3G) решението е скоро идеален за големи слики, додека за мали слики релативната грешка изнесува до 4%.

Моделот каде се користи облаче за процесирање започнува со мала релативна грешка за мали слики и постепено се зголемува до 9% за големи слики. Деталната анализа покажува дека во реалноста мерени се подолги LAN задоцнувања поради зафатеноста на мрежата и поголемите системски времиња за овозможување на одговор од серверот.

Најголема релативна грешка за мали слики дури до 8% е кај моделот со користење на облакот за процесирање, додека најголема релативна грешка за големи слики е кај истото решение и изнесува дури 15%. Ова се случува поради тоа што во овој модел не е вклучена зависноста од моменталниот обем и загушување на мрежата. Комуникациските и пресметувачките перформанси вообичаено се пониски од декларираниите поради оптовареноста на мрежата и искористеноста на целиот систем. Мерењата на LAN задоцнувањето и одговорот од облакот покажуваат вредности кои се повисоки од очекуваните.

8.6 Заклучоци од теоретскиот пристап за разгледување на перформансите

После анализата и обработката на резултатите добиени од теоретските и експерименталните испитувања може да се заклучи дека теоретскиот пристап дава слични резултати како и оние добиени од експериментите. Од добиените резултати може да се види дека, независно која големина на слика се користи за процесирање, убедливо најдобар пристап е кога се користи облачето. Пристапите кога се користи магла или облак за процесирање имаат слични резултатите но се далеку полоши споредбено со пристапот кога се користи облачето. Времињата на процесирање на сликата и OCR про-

цесирање заедно, се за скоро двојно поголеми од тие кога се користи облачето. Убедливо најлош пристап е со користење на мобилниот уред за процесирање. Тука времињата на процесирање кога се користат мали слики се слични со оние кога се користи облак или магла, додека кога се користат средни и големи слики резултатите се многу полоши.

Глава 9

Дискусија и заклучни согледувања

Во докторската дисертација е предложен модел на примена на различни архитектурни решенија за надворешно пресметување во облак, облаче, магла и роса на мобилна апликација која користи препознавање знаци. Претставено е однесувањето на перформансите на процесирање на слика и OCR процесирање, користејќи различни архитектурни решенија кои вклучуваат користење на мобилнен уред, облаче, облак како и пристапот со пресметување во магла. Покрај тоа, направена е теоретска анализа за да се најде влијанието на основните параметри на сликата (број на карактери, големина, комплексност, различност, итн.), како и брзината на мрежата врз процесирање на сликата. Анализата покажа дека коефициентот кој го дефинира трендот на зголемување на ефикасноста на перформансите зависи од задоцнувањето во мрежата и протокот на податоци. Користејќи ја денешната технологија, сценариото каде се користи само мобилниот уред за процесирање ја користи највисоката вредност од овој коефициент додека сценариото со користење на облаче за процесирање ја користи најниската. Ова е експериментално докажано со спроведување на низа експерименти со ново развиените методологии базирани на различни тестни случаи со различен сет на податоци, различни фонтови и комплексност на сликата.

9.1 Споредба со други резултати

Споредбата со други истражувања дава за право да се заклучи дека пристапот претставен во докторската дисертација дава слични резултати, а во повеќето случаи и подобри. Експериментите спроведени во дисертацијата се доволно детални и прецизни за да може да се докаже дека од сите претставени решенија, пристапот со користење на облаче за процесирање на сликата и препознавање на карактери дава најдобри резултати. Исто така моделирањето на перформансите овозможува целосно анализирање на перформансите на пристапите и овозможува полесно одредување на најдоброто решение.

Garcia и останатите во [29] презентираат дека различното време на задоцнување влијае на резултатот, истото тоа е докажано и во докторската дисертација. Исто така, во

докторската дисертација се доаѓа до заклучок дека колку е помал бројот на компоненти кои што се вклучени во архитектурата толку е помало задоцнувањето и побрз протокот на податоци. Ваквиот заклучок е сличен со оној добиен во истражувањето на Caushaj и останатите во [30].

Во многу од решенијата процесирањето се извршува само на еден сервер, или е тоа облакот, или облачето или пак процесирањето се прави само на позадински сервери. За разлика од [36] каде процесирањето се прави само на позадински сервери, решението од докторската дисертација е чекор понапред и може да го користи и мобилниот уред за процесирање. Ваквото решение ги надминува предходните пристапи бидејќи овозможува поширок спектар на архитектури за процесирање, што овозможува да се добијат поточни и подетални резултати

9.2 Осврт на постигнати резултати

Со оглед на тоа што главната хипотеза на ова истражување гласи "Употребата на архитектура за пресметување во облаче и пресметување во магла може во голема мера да го подобри извршувањето на мобилни апликации во споредба со користење на архитектура за пресметување во облак или извршување на апликацијата на мобилниот телефон", од евалуацијата на постигнатите резултати, може да се заклучи дека е целосно исполнета. За препознавање на карактери при користење на мал фонот и мали слики, најдобро решение, анализирајќи ги перформансите само на времето за одговор е, користењето само на мобилниот уред, при што се трошат ресурсите на мобилниот уред и се зголемува потрошувачката на батеријата. Кога сликите се комплексни или се поголеми, секогаш е најдобро процесирањето да се извршува на најблиското облаче или на оддалечен облак што ги потврдува дефинираните хипотези H1 и H2. Исто така, примената на пресметувањето во облаче и пресметувањето во магла резултира со помала искористеност на ресурсите на мобилниот уред на апликацијата за препознавање знаци во однос на другите разгледувани архитектури и решенија што ја потврдува хипотезата H3.

Користењето на слики со мала и средна големина дава најдобри резултати кај решението со користење на мобилен уред за процесирање, додека пристапот со користење на облаче и магла дава најдобри резултати кога се користат слики со големина над 75KB. Пристапот со користење на облак е подобар од мобилното решение кога големината на сликата е поголема од 150KB [R1]. Анализата во која беа опфатени сите архитектурни решенија, покажа дека процесирањето зависи од големината на сликата, но не и од бројот на карактери [R2]. Комплексноста на сликата игра главна улога во процесирањето на сликите. Во некои од случаите процесирачките времиња се скоро идентични. Но, во голем дел од случаите, кога се користат заматени или закосени слики, времето на процесирање е многу поголемо, особено ако се користат слики со голем број на карактери [R3]. Точноста е многу поголема кај слики кои содржат помал број на карактери за препознавање и обратно, додека кај сликите со поголем број на

карактери за препознавање точноста е многу помала. Закосените слики имаат помала точност, додека заматените слики достигнуваат скоро двојно помала точност од оние со нормална комплексност [R4]. Предложената стратегија за одбирање на оптималното решение е базирана на експерименталните резултати добиени користејќи го мобилниот уред, облачето во близина на мобилниот уред, и облакот сместен во Орегон (USA). Од анализата може да се заклучи дека користењето на облаче и магла (3G) како архитектурни решенија, имаат најмало задоцнување и поради тоа се добива набрзо процесирање на сликите [R5]. Според сите анализи може да се заклучи дека целосно се потврдуваат предходно зададените хипотези односно дека најдобар пристап е оној кога се користи облаче за процесирање [R6]. Потрошувачката на батерија е многу поголема кога се користат слики со големина над 1MB. Споредбено со сликите со големина од 500KB или помали, потрошувачката на енергија е за двојно поголема [R7]. Анализата на моделирањето на перформансите на алгоритмот од карактеристиките на сликата покажа дека времето на одговор не зависи од бројот на карактери [R8]. Моделирањето на перформансите на системот за различни архитектурни решенија е дополнување на [R8] и дава одговор на најдобриот архитектурен пристап (облачето) од сите обработени [R9].

9.3 Заклучок

Во текот на изработката на оваа дисертација, користени се достигнувања и публикувани резултати на светски познати автори, но и податоци добиени со континуирано пратење на научната мисла на ова поле. Со користење на различни архитектурни решенија во облак, облаче, магла или роса, предложен е модел за однесување на времето на процесирање на мобилна апликација којашто користи алгоритам за оптичко препознавање знаци, зборови и фрази и даден е одговор кое решение е оптимално.

Од низата направени теоретски истражувања, реализирани експерименти и моделирање на однесувањето времето на процесирање на мобилната апликација, беше констатирано дека:

- употребата на архитектура за пресметување во облаче и пресметување во магла може во голема мера да го подобри извршувањето на мобилните апликации во споредба со користење на архитектура за пресметување во облак или извршување на апликацијата на мобилен уред;
- примената на пресметувањето во облаче и пресметувањето во магла дозволува побрз трансфер на податоци и побрзо процесирање на апликацијата за препознавање карактери, во однос на другите разгледувани архитектурни решенија;
- примената на пресметувањето во облаче и пресметувањето во магла резултира со помала искористеност на ресурсите на мобилниот уред на апликацијата за препознавање знаци во однос на другите разгледувани архитектурни решенија;
- употребата на пристапот за пресметување во облаче и пресметување во магла овозможува помало искористување на ресурсите на мобилниот уред;

Во случај на слики со мала и средна големина, најдобри резултати се постигнуваат кога се користи процесирање со мобилен уред, додека пристапите со облаче и магла обезбедуваат најдобри резултати кога се користат за слики од над 75КВ. Пристапот за пресметување во облаче е особено подобар од пристапот со мобилен уред кога големината на сликата е поголема од 150КВ. Анализата во која се опфатени сите архитектурни решенија, покажа дека обработката зависи од големината на сликата, но не и од бројот на карактери.

Исто така, беше заклучено дека комплексноста на сликата игра голема улога врз времето за процесирање на сликата. Во случај кога се користат заматени или закосени слики, времето за процесирање е многу поголемо, особено ако се користат слики со голем број карактери.

Точноста е многу повисока кај слики кои содржат помал број карактери за препознавање. Точноста на сликите со поголем број карактери е многу помала. Закосените слики имаат помала прецизност, додека заматените слики достигнуваат речиси половина од точноста на оние со нормална комплексност.

Предложената стратегија за избор на оптимално решение беше заснована на експерименталните резултати добиени со помош на мобилниот уред, облачето во близина на мобилниот уред и облакот лоциран во Орегон (САД). Од анализата може да се заклучи дека употребата на пресметки во облаче и пресметки во магла (3G) како архитектурни решенија има најмало задоцнување, и затоа е можно брзо процесирање на сликите.

Потрошувачката на батеријата е многу поголема кога се користат слики со големина од 1МВ. Во споредба со сликите од 500КВ или помали, потрошувачката на енергија е двапати поголема.

Анализата на моделирањето на однесувањето на времето на процесирање, во зависност од карактеристиките на сликата покажа дека и во овој случај, како и во експерименталните истражувања, времето на одговор не зависи од бројот на карактери. Моделирањето на однесувањето на времето на процесирање за различни архитектурни решенија даде потврда на прашањето за најдобар архитектурен пристап, а тоа е **облачето**.

9.4 Придобивки од добиените модели

Постапката за валидација на развиениот модел дава одговор на истражувачкото прашање и овозможува избор на најдобро решение за OCR процесирање.

Ако се земе во предвид дека пристапот со користење на магла (4G) сеуште е премногу скап и телекомуникациските оператори немаат некои бизнис интереси за имплементација на OCR базирани сервиси, може да се заклучи дека пристапот со користење на облаче е најдобар во споредба со останатите анализирани пристапи.

Предноста на пристапот со користење на облаче е во тоа што може користејќи Wi-Fi поврзување да се прави процесирање на слики и OCR процесирање на локации кои што немаат пристап до Интернет или пак истиот е многу скап (Интернет во роаминг). Како пример може да ја земеме употребата на ваквиот пристап во трговски центар

во земја каде што не го знаеме јазикот (на пример Кина) и сепак сакаме да дознаеме што пишува. Трговскиот центар може да нуди Wi-Fi поврзување за поврзување до мал сервер или пак може да користиме 3G/4G поврзување во роадинг или пак да имаме инсталирано специјален сервер кој нуди микро сервиси за преведување. Земајќи во предвид дека цената за користење на 3G/4G во роадинг е многу голема, останува пристапот со користење на облаче за процесирање (поврзување преку Wi-Fi) убедливо најдобро решение.

Земајќи ги во предвид доцнењата и проблемите кои произлегуваат од користењето на пристапот со користење на облак, решението со користење на облаче би имал најефикасна примена. Посебно би бил корисен за туристи на познати туристички места кои се задржуваат кратко. Сите тие би ги користеле локалните облачиња за процесирање на слики со знаци, текстови, наслови, и добивање на соодветно значење на истите назад на мобилниот уред. Ова претставува погоден бизнис модел за малите и средни бизниси кои преку иновативни решенија би направиле диверзификација на нивната понуда.

Предложената методологија за одбирање на оптималното решение се заснова на теоретска анализа и на експерименталните резултати добиени со користење на мобилниот уред, облачето и облакот како процесирачки единици. Од сите направени експерименти и добиени резултати лесно е да се заклучи дека најдобри резултати се постигнуваат кога се користи облачето како пристап за процесирање на сликите и OCR процесирање.

Препознавањето на карактери користејќи го OCR алгоритмот, во случај кога имаме кратки текстови и мали слики најдобро е да се користи пристапот со процесирање на самиот мобилен уред. Доколку се користат комплексни и големи слики, тогаш е подобро процесирањето да се извршува на најблискиот сервер (облаче) или на оддалечен сервер (облак).

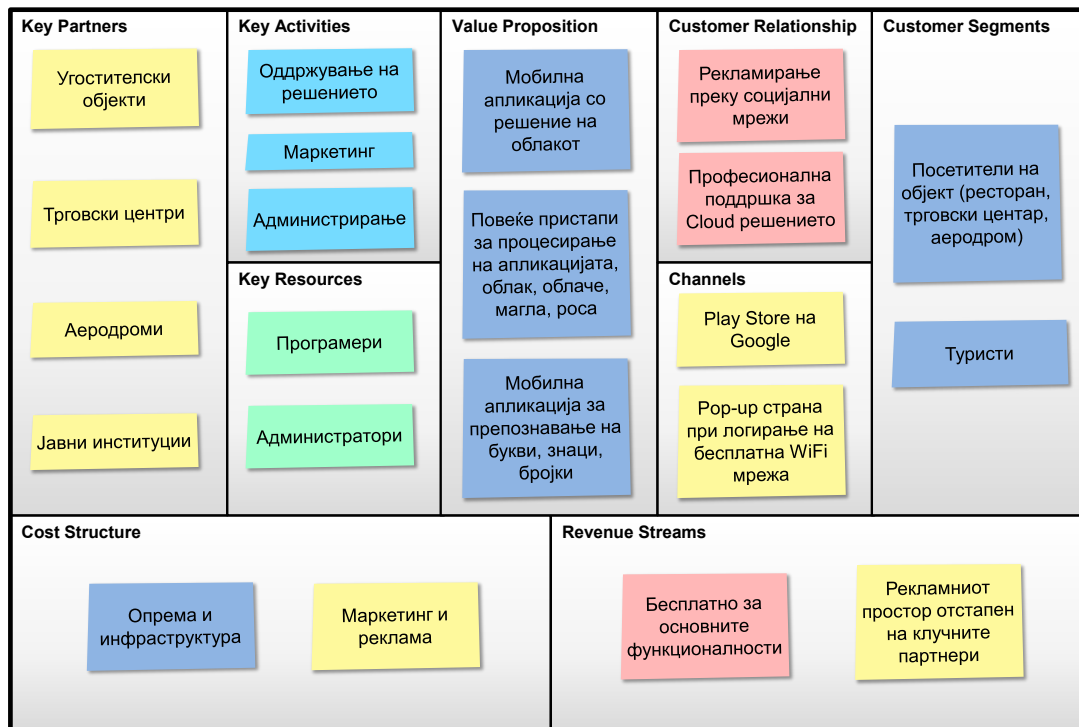
Со забрзаниот развој на 5G мрежната технологија, користењето на апликациите кои се извршуваат во реално време би овозможило нивно побрзо извршување на истите. Знаејќи дека брзината е клучна при пренесување на сликата кон позадинските сервери (облак, облаче, магла), користењето на оваа мрежна технологија ќе доведе до побрзо и поефикасно процесирање на сликата.

Решението кое се истражуваше и развиваше е основа за мобилен OCR преведувач, чиј што бизнис модел, ако се комерцијализира, е прикажан на Слика 9.1.

9.5 Идни истражувања

Идните истражувања ќе бидат во насока на тоа точно да се одреди методологија кога и дали има потреба од пренесување на процесирањето и користење на пристапите за пресметување во облаче, магла или облак. Исто така постојат можности за надоградување на бизнис моделот, за потребите на телекомуникациските оператори за користење на архитектурните пристапи со користење на магла (4G) а подоцна и на магла (5G), што би довело до зголемување на задоволството на нивните корисници и намалување

Mobile OCR Translator



Слика 9.1: Canvas бизнис модел

на искористеноста на ресурсите на нивните мобилни уреди. Може да се каже дека ова е почеток на една нова ера во мобилното пресметување, којшто би овозможил целосно пренесување на процесирањето на мобилните апликации, посебно на оние кои побаруваат големи ресурси за нивно извршување.

Библиографија

- [1] Aleksandar Bahtovski and Marjan Gušev. Analysis of cloud portability. 2013.
- [2] Aleksandar Bahtovski and Marjan Gusev. Interoperability of cloud and mobile services.
- [3] Mahadev Satyanarayanan, Paramvir Bahl, Ramón Caceres, and Nigel Davies. The case for vm-based cloudlets in mobile computing. *IEEE pervasive Computing*, 8(4), 2009.
- [4] Karolj Skala, Davor Davidovic, Enis Afgan, Ivan Sovic, and Zorislav Sojat. Scalable distributed computing hierarchy: Cloud, fog and dew computing. *Open Journal of Cloud Computing (OJCC)*, 2(1):16–24, 2015.
- [5] Aleksandar Bahtovski and Marjan Gušev. Cloud computing in mobile technologies. 2012.
- [6] Karthik Kumar and Yung-Hsiang Lu. Cloud computing for mobile users: Can offloading computation save energy? *Computer*, 43(4):51–56, 2010.
- [7] Emmanouil Koukoumidis, Dimitrios Lymberopoulos, Karin Strauss, Jie Liu, and Doug Burger. Pocket cloudlets. In *ACM SIGPLAN Notices*, volume 46, pages 171–184. ACM, 2011.
- [8] Aleksandar Bahtovski and Marjan Gusev. Cloudlet challenges. *Procedia Engineering*, 69:704–711, 2014.
- [9] Tolga Soyata, Rajani Muraleedharan, Colin Funai, Minseok Kwon, and Wendi Heinzelman. Cloud-vision: Real-time face recognition using a mobile-cloudlet-cloud acceleration architecture. In *Computers and Communications (ISCC), 2012 IEEE Symposium on*, pages 000059–000066. IEEE, 2012.
- [10] Aleksandar Bahtovski and Marjan Gusev. Multilingual cloudlet-based dictionary. In *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on*, pages 380–385. IEEE, 2014.
- [11] Aleksandar Bahtovski and Marjan Gusev. Developing challenges for a cloudlet solution of a multi-lingual dictionary application. In *Computational Intelligence, Communication Systems and Networks (CICSyN), 2014 Sixth International Conference on*, pages 125–130. IEEE, 2014.

- [12] Aleksandar Bahtovski, Katerina Zdravkova, and Marjan Gusev. Performance of cloudlet-based multilingual dictionary. In *Proceedings of the 7th Balkan Conference on Informatics Conference*, page 33. ACM, 2015.
- [13] Sasko Ristov, Kiril Cvetkov, and Marjan Gusev. Implementation of a horizontal scalable balancer for dew computing services. *Scalable Computing: Practice and Experience*, 17(2):79–90, 2016.
- [14] Ivan Stojmenovic and Sheng Wen. The fog computing paradigm: Scenarios and security issues. In *Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on*, pages 1–8. IEEE, 2014.
- [15] Tom H Luan, Longxiang Gao, Zhi Li, Yang Xiang, Guiyi Wei, and Limin Sun. Fog computing: Focusing on mobile users at the edge. *arXiv preprint arXiv:1502.01815*, 2015.
- [16] Masashi Koga, Ryuji Mine, Tatsuya Kameyama, Toshikazu Takahashi, Masahiro Yamazaki, and Teruyuki Yamaguchi. Camera-based kanji ocr for mobile-phones: Practical issues. In *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, pages 635–639. IEEE, 2005.
- [17] KS Bae, KK Kim, YG Chung, and WP Yu. Character recognition system for cellular phone with camera. In *Computer Software and Applications Conference, 2005. COMPSAC 2005. 29th Annual International*, volume 1, pages 539–544. IEEE, 2005.
- [18] Shuji Senda, Kyosuke Nishiyama, Toshiyuki Asahi, and Keiji Yamada. Camera-typing interface for ubiquitous information services. In *Pervasive Computing and Communications, 2004. PerCom 2004. Proceedings of the Second IEEE Annual Conference on*, pages 366–369. IEEE, 2004.
- [19] Mikael Laine and Olli S Nevalainen. A standalone ocr system for mobile cameraphones. In *Personal, Indoor and Mobile Radio Communications, 2006 IEEE 17th International Symposium on*, pages 1–5. IEEE, 2006.
- [20] GOCR. Gocr, June 2000.
- [21] OCRAD. Ocrad, January 2016.
- [22] Thomas M Breuel. The ocrpus open source ocr system. In *Electronic Imaging 2008*, pages 68150F–68150F. International Society for Optics and Photonics, 2008.
- [23] OpenOCR. Openocr, March 2016.
- [24] Ray Smith. An overview of the tesseract ocr engine. In *Proc. Ninth Int. Conference on Document Analysis and Recognition (ICDAR)*, pages 629–633. IEEE, 2007.

- [25] Anand Joshi, Mi Zhang, Ritesh Kadmawala, Karthik Dantu, Sameera Poduri, and Gaurav Sukhatme. Ocrdroid: A framework to digitize text using mobile phones. In *Proc. ICST International Conference on Mobile Computing, Applications, and Services*. Citeseer, 2009.
- [26] ABBYY. Abbyy, March 2016.
- [27] OmniPage. Omnipage, March 2016.
- [28] ExperVision. Expervision, March 2016.
- [29] Johan Garcia, Stefan Alfredsson, and Anna Brunstrom. Delay metrics and delay characteristics: A study of four swedish hsdpa+ and lte networks. In *Networks and Communications (EuCNC), 2015 European Conference on*, pages 234–238. IEEE, 2015.
- [30] Eralda Caushaj, Ivan Ivanov, Huirong Fu, Ishwar Sethi, and Ye Zhu. Evaluating throughput and delay in 3g and 4g mobile architectures. *Journal of Computer and Communications*, 2014, 2014.
- [31] A-Long Jin, Wei Song, Ping Wang, Dusit Niyato, and Peijian Ju. Auction mechanisms toward efficient resource sharing for cloudlets in mobile cloud computing. *IEEE Transactions on Services Computing*, 9(6):895–909, 2016.
- [32] Ambrose A Azeta, Nicholas A Omoregbe, Sanjay Misra, Adewole Adewumi, and Temitope O Olokunde. Adapted cloudlet for mobile distance learning: Design, prototype and evaluation. In *ICADIWT*, pages 220–228, 2016.
- [33] Yu Cao, Songqing Chen, Peng Hou, and D. Brown. Fast: A fog computing assisted distributed analytics system to monitor fall for stroke mitigation. In *2015 IEEE International Conference on Networking, Architecture and Storage (NAS)*, pages 2–11, Aug 2015.
- [34] Vladimir Stantchev, Ahmed Barnawi, Sarfaraz Ghulam, Johannes Schubert, and Gerrit Tamm. Smart items, fog and cloud computing as enablers of servitization in healthcare. *Sensors & Transducers*, 185(2):121, 2015.
- [35] Yingwei Wang. Cloud-dew architecture. *International Journal of Cloud Computing*, 4(3):199–210, 2015.
- [36] Nitin Mishra and C Patvardhan. Atma: Android travel mate application. *International Journal of Computer Applications*, 50(16), 2012.
- [37] worldens. worldens, March 2016.
- [38] Victor Fragoso, Steffen Gauglitz, Shane Zamora, Jim Kleban, and Matthew Turk. Translatar: A mobile augmented reality translator. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 497–502. IEEE, 2011.

- [39] Keke Gai, Meikang Qiu, Hui Zhao, Lixin Tao, and Ziliang Zong. Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing. *Journal of Network and Computer Applications*, 59:46–54, 2016.
- [40] Cheng Feng, Wendong Wang, Ye Tian, Xirong Que, and Xiangyang Gong. Estimate air quality based on mobile crowd sensing and big data. In *A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2017 IEEE 18th International Symposium on*, pages 1–9. IEEE, 2017.
- [41] John K Zao, Tchin Tze Gan, Chun Kai You, Sergio José Rodríguez Méndez, Cheng En Chung, Yu Te Wang, Tim Mullen, and Tzyy Ping Jung. Augmented brain computer interaction based on fog computing and linked data. In *Intelligent Environments (IE), 2014 International Conference on*, pages 374–377. IEEE, 2014.
- [42] Bhaviya Rajesh Gandani, Ami Rajesh Gandani, Kushal Dharmesh Doshi, and Ankur Naishadh Gandhi. Unicode optical character recognition using neural networks.
- [43] Dipti Singh, Atul Bansal, and Neha Bansal. A review on optical character recognition using various techniques.
- [44] Language Technologies Unit (Canolfan Bedwyr). An overview of the tesseract ocr (optical character recognition) engine, and its possible enhancement for use in Wales in a pre-competitive research stage, April 2008.
- [45] Yiduo Mei, Ling Liu, Xing Pu, and Sankaran Sivathanu. Performance measurements and analysis of network i/o applications in virtualized cloud. In *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, pages 59–66. IEEE, 2010.
- [46] Liu XiaoFang, Chang Chi-Yuan, and Roshanravan Rouzbeh. Simulation comparison between lte and wi-fi in networks.
- [47] Apostolos Papageorgiou, Ehsan Poormohammady, and Bin Cheng. Edge-computing-aware deployment of stream processing tasks based on topology-external information: Model, algorithms, and a storm-based prototype. In *Big Data (BigData Congress), 2016 IEEE International Congress on*, pages 259–266. IEEE, 2016.
- [48] Mahadev Satyanarayanan. The emergence of edge computing. *Computer*, 50(1):30–39, 2017.
- [49] Pieter Simoens, Yu Xiao, Padmanabhan Pillai, Zhuo Chen, Kiryong Ha, and Mahadev Satyanarayanan. Scalable crowd-sourcing of video from mobile devices. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 139–152. ACM, 2013.
- [50] Yaser Jararweh, Ahmad Doulat, Omar AlQudah, Ejaz Ahmed, Mahmoud Al-Ayyoub, and Elhadj Benkhelifa. The future of mobile cloud computing: Integrating cloudlets

- and mobile edge computing. In *Telecommunications (ICT), 2016 23rd International Conference on*, pages 1–5. IEEE, 2016.
- [51] Zhanghui Liu, Xuee Zeng, Wensi Huang, Junxin Lin, Xing Chen, and Wenzhong Guo. Framework for context-aware computation offloading in mobile cloud computing. In *Parallel and Distributed Computing (ISPDC), 2016 15th International Symposium on*, pages 172–177. IEEE, 2016.
- [52] TechTarget. Ieee 802 wireless standards: Fast reference, 2006.
- [53] PassMark® Software. Cpu benchmarks, 2017.
- [54] Aleksandar Bahtovski and Marjan Gusev. Performance evaluation and modeling of a cloud-based mobile OCR solution. Technical Report TR-15/2017, Sts. Cyril and Methodius University, Faculty of Computer Science and Engineering, Skopje, Macedonia, 2017.