

# Тема: ViZTranZ - превод на обекти от изображение

Предмет: Приложно-програмни интерфейси за работа с облачни архитектури с Амазон Уеб Услуги (AWS)

Изготвил: Симеон Христов, фн: 71845, имейл: s.e.hristov99@gmail.com

## Съдържание

<b>1</b>	<b>Условие</b>	<b>3</b>
<b>2</b>	<b>Въведение</b>	<b>3</b>
<b>3</b>	<b>Теория</b>	<b>4</b>
3.1	Работа в режим AWS . . . . .	4
3.2	Работа в режим Local . . . . .	4
<b>4</b>	<b>Използвани технологии</b>	<b>5</b>
4.1	Технологии, свързани с AWS . . . . .	5
4.2	Други технологии . . . . .	5
<b>5</b>	<b>Инсталация и настройки</b>	<b>6</b>
<b>6</b>	<b>Кратко ръководство за потребителя и примерни данни</b>	<b>6</b>
<b>7</b>	<b>Описание на програмния код</b>	<b>10</b>
7.1	Файл app.py . . . . .	10
7.2	Файл constants.py . . . . .	10
7.3	Файл helper.py . . . . .	11
7.4	Файл lambda.py . . . . .	11
7.5	Файл s3-manager.py . . . . .	11
7.6	Файл tf-od.py . . . . .	11
7.7	Файл translation.py . . . . .	12
<b>8</b>	<b>Приноси на студента, ограничения и възможности за бъдещо развитие</b>	<b>12</b>

9	Какво научих	12
10	Списък с фигури	13
11	Използвани източници	13

# 1 Условие

ViZTranZ е уеб приложение, което позволява на потребител да прикачи снимка и да получи превод на обектите, разпознати в тази снимка.

## 2 Въведение

Приема се, че преводът винаги се случва от английски към някой от езиците български, немски и руски. Превеждат се етикетите на първите 10 разпознати обекта с най-голяма увереност. Възможно е да не бъдат разпознати обекти в снимката, при което се извежда подходящо съобщение на потребителя.

В странично (ляво) меню се предлагат двата начина на работа с приложението - с използване услугите на AWS (наричан в приложението "AWS") и без използване услугите на AWS (наричан в приложението "Local"). По подразбиране е избран "Local".

Начинът на работа се описва с тези стъпки:

1. Система показва меню, от което може да бъде прикачена снимка.
2. Потребителят прикачва (точно една) снимка.
3. Система показва меню, от което може да бъдат избрани езиците, към които да бъдат преведени разпознатите обекти. Поддържани езици са български, немски и руски. По подразбиране е избран български. Необходимо е да бъде избран поне един език, за да продължи процесът.
4. Потребителят избира поне един език.
5. Системата показва бутон Translate, след натискането на който започва процесът по разпознаване на обектите от снимката и преводът на тяхните етикети.
6. Потребителят натиска появилия се (най-отдолу) бутон.
7. В зависимост от начина на работа на приложението се преминава през процеса на разпознаване на обектите от снимката и техния превод.
8. Системата получава резултатите от превода на етикетите на разпознатите обекти (за всички езици).
9. Системата показва на потребителя графика (тип бар диаграма), в която по абсцисата са нанесени вероятностите за съществуването на разпознатите обекти в снимката, а по ординатата - преводът на обектите, заедно с етикета на английски.

10. Потребителят може да избере различни езици, при което резултати ще се изведат мигновено, т.к. те са налични в паметта, отделена за програмата. По този начин се избягва повторно качване на снимката и задействане на процеса.

## 3 Теория

В зависимост от начина на работа тук са разгледани техническите детайли по време на процеса по разпознаване на обекти и превод на техните етикети.

### 3.1 Работа в режим AWS

Когато потребителят натисне бутона Translate, снимката бива качена в предварително създадена база от данни, реализирана в S3. При всяко качване на обект в тази база се задейства чрез тригер ламбда функция. В нея се извлича най-скорошно качената снимка. Тя се подава на Rekognition, откъдето се получава речник с информация за разпознатите обекти. От този речник се извличат етикетите на разпознатите обекти и увереността на модела, че те съществуват в снимката. Всеки етикет (поотделно) се предава на Translate и за трите езика. Преводите се конкатенират със списъка от един елемент - увереността. По този начин се получава речник с ключове - символни низове, представляващи етикетите на разпознатите обекти (на английски език), и стойности - хетерогенни списъци, чиито първи елементи са увереностите на модела, а следващите три са символни низове, представляващи превода на ключа в трите езика. Този речник се записва в друга база от данни, отново в S3, откъдето е (поименно) достъпена в приложението. Резултатите се съхраняват под формата на json. След изтегляне в приложението резултатите се трансформират в таблица (pandas dataframe) и се визуализират под формата на bar диаграма, създадена чрез plotly.

### 3.2 Работа в режим Local

В този режим се използват две основни технологии - Tensorflow и Hugging Face Transformers. Tensorflow е рамка за създаване на невронни мрежи. В случая се използва вече създадена и тренирана невронна мрежа за разпознаване на обекти в снимки - mobilenetv2, чиито тегла предварително са изтеглят по време на процесът по подготвяне на приложението. От нея отново се взимат разпознатите етикети и увереността в тях. С цел бързодействие избраният модел не е толкова добър в разпознаването на обекти и поради тази причина се наблюдават по-слаби резултати. Понеже няма модел (невронна мрежа), който е достатъчно малък и бърз, за превод от английски към трите езика на веднъж, етикетите

се превеждат чрез три модела. Моделът opus-mt-en-bg се използва за превод от английски към български. Моделът opus-mt-en-ru се използва за превод от английски към руски. Моделът opus-mt-en-de се използва за превод от английски към български.

## 4 Използвани технологии

### 4.1 Технологии, свързани с AWS

- **AWS Identity and Access Management (IAM)** - за създаване на роли за ламбда функцията, както и за достъпът до S3.
- **AWS Cost Explorer** - за създаване на бюджет и сигнализация при надминаването му.
- **Amazon Simple Storage Service (Amazon S3)** - за съхраняване на снимките и резултатите.
- **AWS Lambda** - за мост между S3, Rekognition, и Translate.
- **AWS Rekognition** - за разпознаване на обекти.
- **AWS Translate** - за превод етикетите на обекти.
- **AWS CloudWatch** - за наблюдаване на работата на ламбда функцията.

### 4.2 Други технологии

- **Python** - езикът, на който е реализирано приложението.
- **Streamlit** - библиотека на Python, чрез която е създадено уеб приложението.
- **Tensorflow** - за създаване и използване на невронни мрежи. В случая, използвана за разпознаване на обекти от изображение.
- **Hugging Face** - за работа с текст. В случая, използвана за превод етикетите на обекти.

## 5 Инсталация и настройки

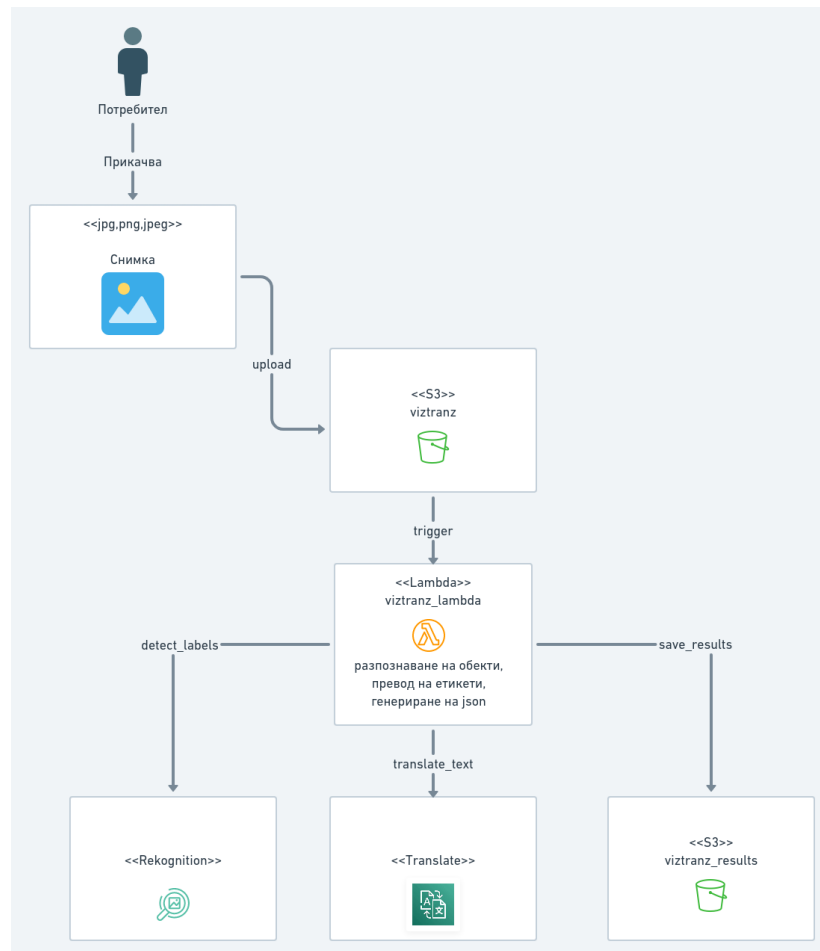
Стъпка 1. Отваря се терминал в директорията на приложението.

Стъпка 2. Пуска се файлът setup.sh.

Стъпка 3. (Незадължителна) Ако ще се използва режим на работа AWS е нужно да се поставят данни за удостоверяване в копие на .env-example.

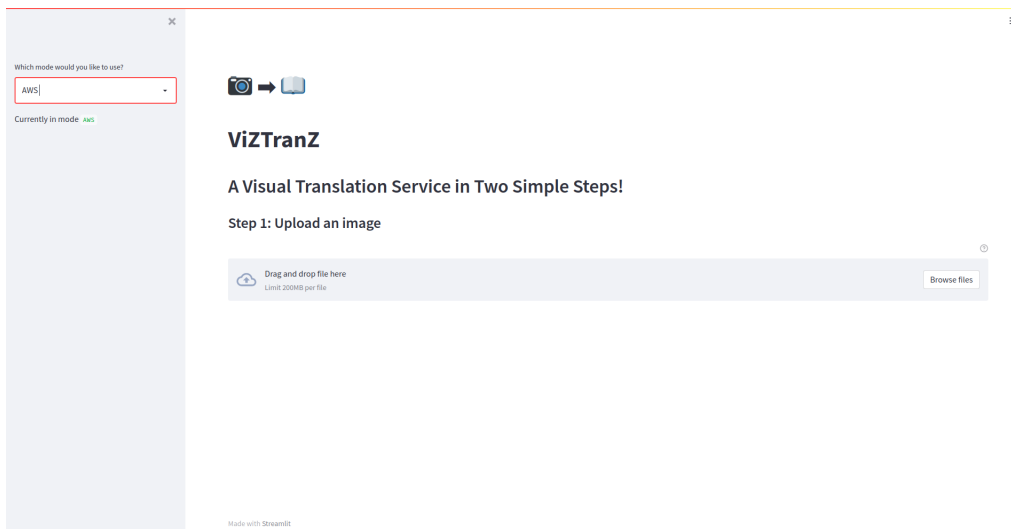
Стъпка 4. Пуска се файлът run.sh.

## 6 Кратко ръководство за потребителя и примерни данни

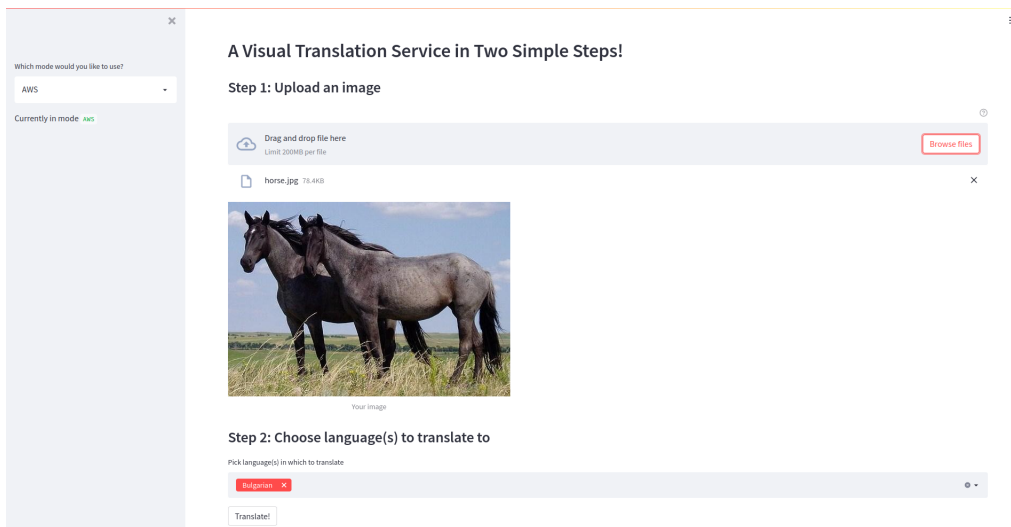


Фигура 1: Структура на услугите, използвани в режим AWS

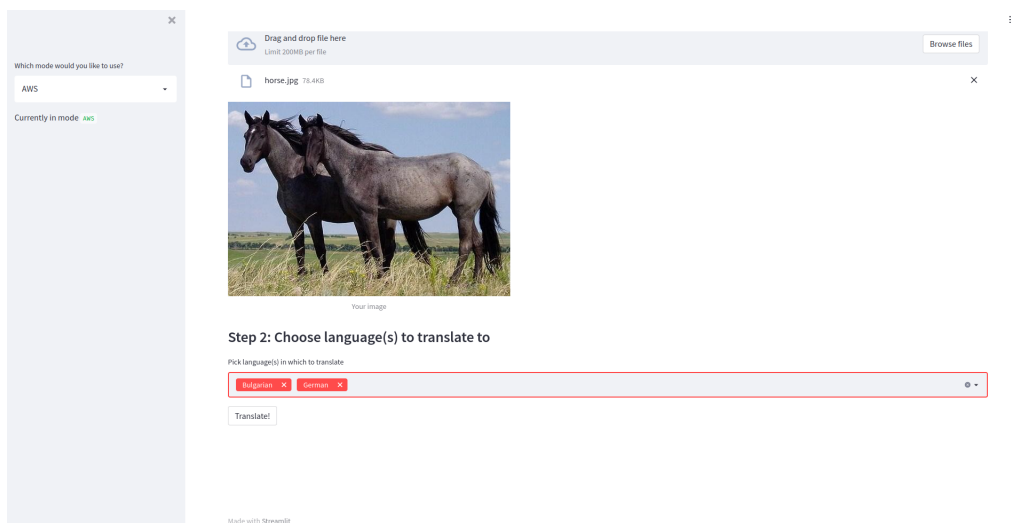
За да се получат правилно преводи е нужно да се качи снимка, на която има поне един ясно различим обект. Следващите стъпки показват примерна ситуация.



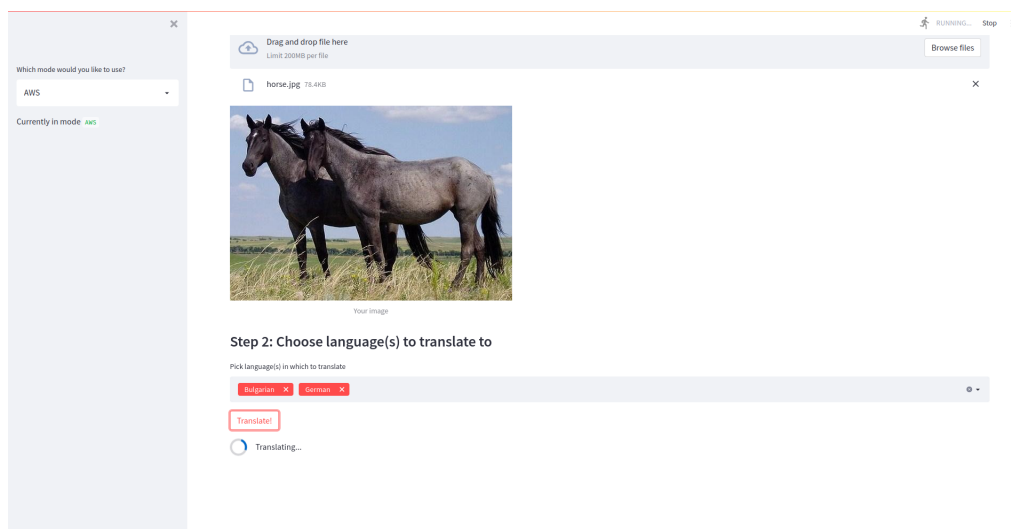
Фигура 2: Стъпка 1. Избира се AWS от ляво меню.



Фигура 3: Стъпка 2. Прикачва се снимка.

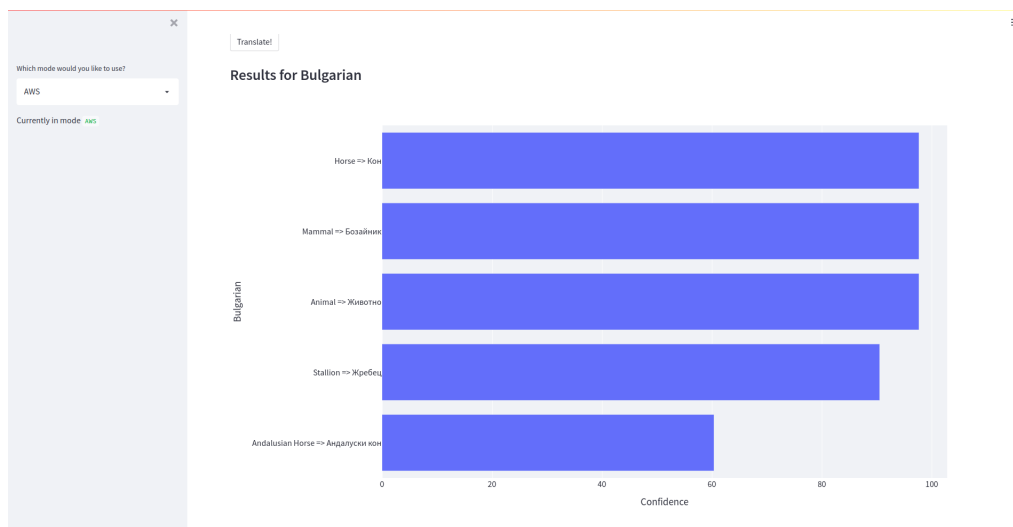


Фигура 4: Стъпка 3. Избират се езици за превод.

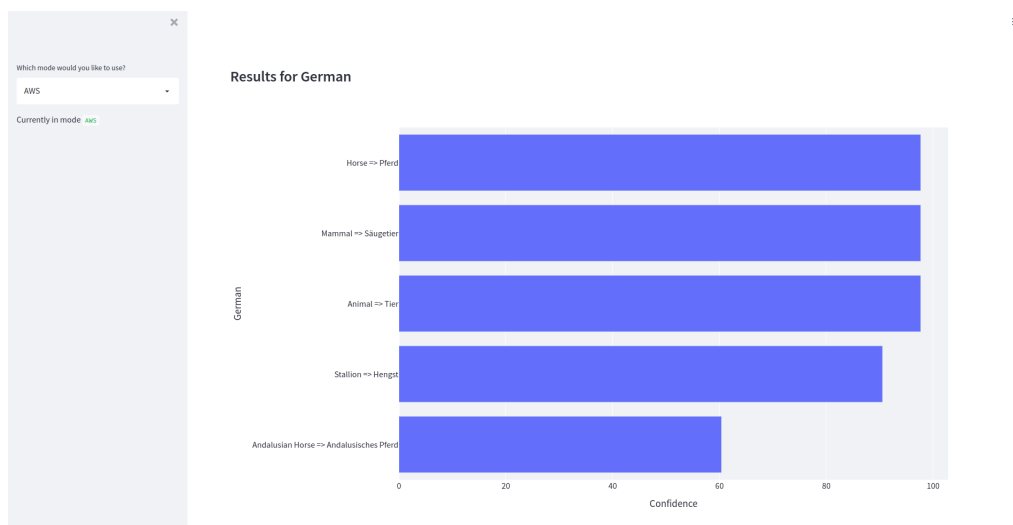


Фигура 5: Стъпка 4. Натиска се бутон Translate.





Фигура 6: Стъпка 5.1. Наблюдават се резултати за български



Фигура 7: Стъпка 5.2. Наблюдават се резултати за немски

## 7 Описание на програмния код

Приложението се състои от един пакет - viztranz и един външен файл (модул), който управлява приложението - app.py.

### 7.1 Файл app.py

Основният файл, който генерира уеб съдържанието и чрез който се управлява цялото приложение. Разделен е на три части - заглавна част, странично меню и главно меню. В заглавната част се поставя настройка за използване на режим широк екран, при който всяко поле (widget) ще заема максимална ширина. Поставя се също и икона на страницата, както и заглавие. Те могат да бъдат видяни в табът, отворен от браузъра. С последните три команди се дава заглавие и уточняващ текст, който описва целта на приложението.

В секцията странично меню се задава selectbox, чрез който потребителят може да избере режим на работа. След промяна на режима на работа, ако са съществували резултати от предишен превод, те ще бъдат изтрети.

В секцията главно меню се показват различните подменюта. Първо, се показва това за прикачване на файл. Може да прикачен точно един файл. Ако това се случи, снимката се визуализира и потребителят е подканен да избере езици, на които желае да преведе разпознатите обекти. За да се продължи напред, е нужно да бъде избран поне един език. Ако това се случи се показва бутон Translate, при натискането на който, в зависимост от режима на работа, се извикват съответните методи за разпознаване на обекти и превод на тяхните етикети. Понеже тези процеси отнемат време, се визуализира и spinner, който гарантира, че приложението работи. Когато бъдат получени резултатите, те се запазват в паметта на програмата и се поставят в pandas dataframe, чрез който лесно могат да се визуализират под формата на хоризонтална bar диаграма.

### 7.2 Файл constants.py

Тук се съхраняват глобални променливи и константи. Константата SUPPORTED-LANGS задава поддържаните от приложението езици за превод. Редът е от значение е нови езици е нужно да бъдат добавяни в края. Променливата results съхранява получените резултати след превод. Чрез нея се избягва повторно действие на процедурата по разпознаване и превод. С други думи, ако потребителят реши да получи други преводи за същата снимка ще се използва тази променлива. Текущият режим на работа се записва в променливата mode. Константата MODULE-HANDLE съхранява пътят до директорията, съхраняваща моделът, използван от Tensorflow за разпознаване на обекти в режим на работа

Local. Функцията `get-sample-results` връща примерни резултати. Би се използвала за тестване, без да е нужно преминаване през AWS.

### 7.3 Файл `helper.py`

Съхранява помощни функции, използвани в основния файл (`app.py`).

Функцията `toggle-results` затрива текущо запазените резултати за превод. Тя се извиква, когато се качи нова снимка или се избере друг режим на работа.

Функцията `get-upl-file` показва меню за прикачване на снимка и връща прикачената снимка, запазена в паметта.

Функцията `get-langs` показва меню за избор на език и връща избраните езици под формата на списък от символни низове.

Функцията `build-chart` създава `pandas dataframe` и го визуализира под формата на `bar` диаграма. По абсцисата се представя увереността на модела, че такъв обект съществува в снимката, а по ординатата - превод на съответния етикет.

Функцията `add-translations` добавя преводи към разпознатите обекти. Използва се в режим Local.

### 7.4 Файл `lambda.py`

Файл, който съхранява кодът на ламбда функцията, дефинирана в AWS.

### 7.5 Файл `s3-manager.py`

Съхранява кодът, използван за работа с S3.

Понеже за свързване с S3 са нужен акаунт, първите неща, които се зареждат в този файл са двата ключа, необходими за оторизация и името на базата от данни, в която ще бъде записан резултатът от превода. След това зареждане се дефинира и клиентът, който ще "общува" с AWS.

Функцията `gen-name` генерира име на файла, който ще бъде записан в базата от данни, която ще задейства ламбда функцията, и база от данни, в която ще бъдат записани резултати. Това е необходимо, т.к. няма достъп до самата ламбда и файловете се разпознават само по тяхното име.

Функцията `upload` прикачва изображението в S3.

Функцията `get-results` извлича резултатите от базата от данни, която ги съхранява.

### 7.6 Файл `tf-od.py`

Тук се съхранява кодът, който извършва разпознаване на обекти в режим на работа Local.

Променливата `detector` съхранява използвания модел.  
Функцията `run-detector` извършва разпознаването на обекти.

## 7.7 Файл `translation.py`

Тук се съхранява кодът, който извършва превод на етикетите на разпознатите обекти в режим на работа `Local`.

Променливите `tr-ru`, `tr-de`, и `tr-bg` са инстанции на трите модела за превод.

Речникът `translators` съхранява описаните по-горе инстанции на класовете.

Функцията `translate` използва даден модел за превод на текст и връща превода.

## 8 Приноси на студента, ограничения и възможности за бъдещо развитие

Създадено е приложение, което може да бъде използвано за превод на обекти от снимки в реално време. Подходящо е за деца, които опознават света, и учат чужди езици.

Към момента има ограничения в точността на моделите в режим `Local`. Възможно е получаване на грешни резултати с тях. Второто ограничение е в поддържаният набор езици.

В бъдеще е възможно да се добавят други езици, както и да се визуализират преводите не в `bar chart`, а директно върху изображението, като се използват ограждащите кутии (`bounding box`), които се връщат от `AWS Rekognition` по време на разпознаването на обекти. Също така може да се използват двете бази от данни в `S3` за анализ на начина, по който се използва приложението (какви снимки се качват, кога има най-много трафик и т.н.) с цел неговото подобряване.

## 9 Какво научих

Научих се как да използвам изброените по-горе услуги, предоставени от `AWS`. Научих се как да ги интегрирам в реално приложение. Усъвършенствах познанията си по `Python` и `Streamlit`.

## 10 Списък с фигури

### Списък на фигурите

1	Структура на услугите, използвани в режим AWS . . . . .	6
2	<b>Стъпка 1.</b> Избира се AWS от ляво меню. . . . .	7
3	<b>Стъпка 2.</b> Прикачва се снимка. . . . .	7
4	<b>Стъпка 3.</b> Избират се езици за превод. . . . .	8
5	<b>Стъпка 4.</b> Натиска се бутон Translate. . . . .	8
6	<b>Стъпка 5.1.</b> Наблюдават се резултати за български . . . . .	9
7	<b>Стъпка 5.2.</b> Наблюдават се резултати за немски . . . . .	9

## 11 Използвани източници

- [1] AWS Lambda Documentation
- [2] Amazon Simple Storage Service Documentation
- [3] Amazon Rekognition Documentation
- [4] Amazon Translate Documentation
- [5] Boto3 documentation
- [6] Streamlit documentation
- [6] Tensorflow documentation
- [7] Tensorflow Hub
- [8] Hugging Face Transformers for Translations: opus-mt-de-bg
- [9] Hugging Face Transformers for Translations: t5-small