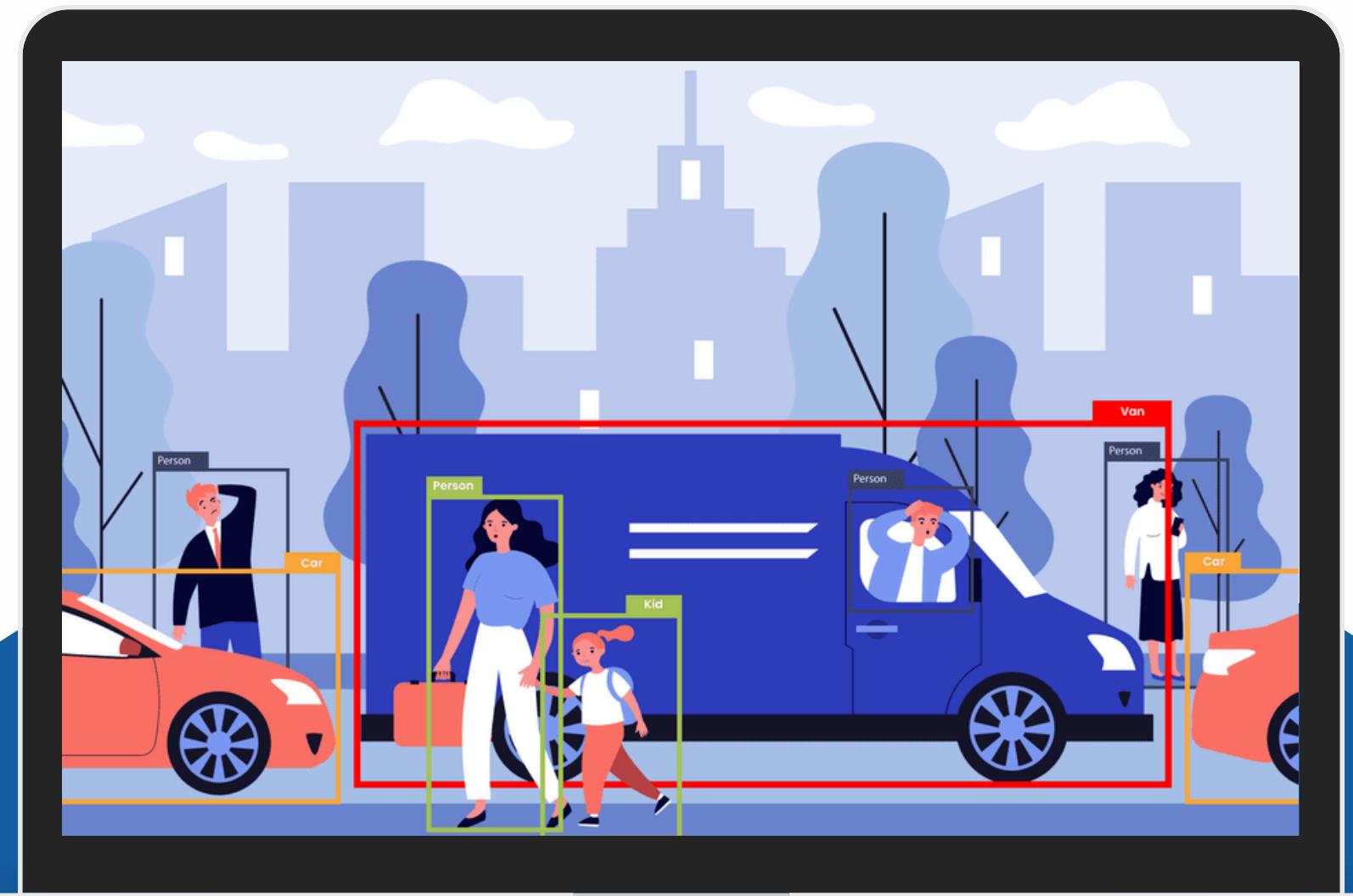


IO

Детекција и броење на возила

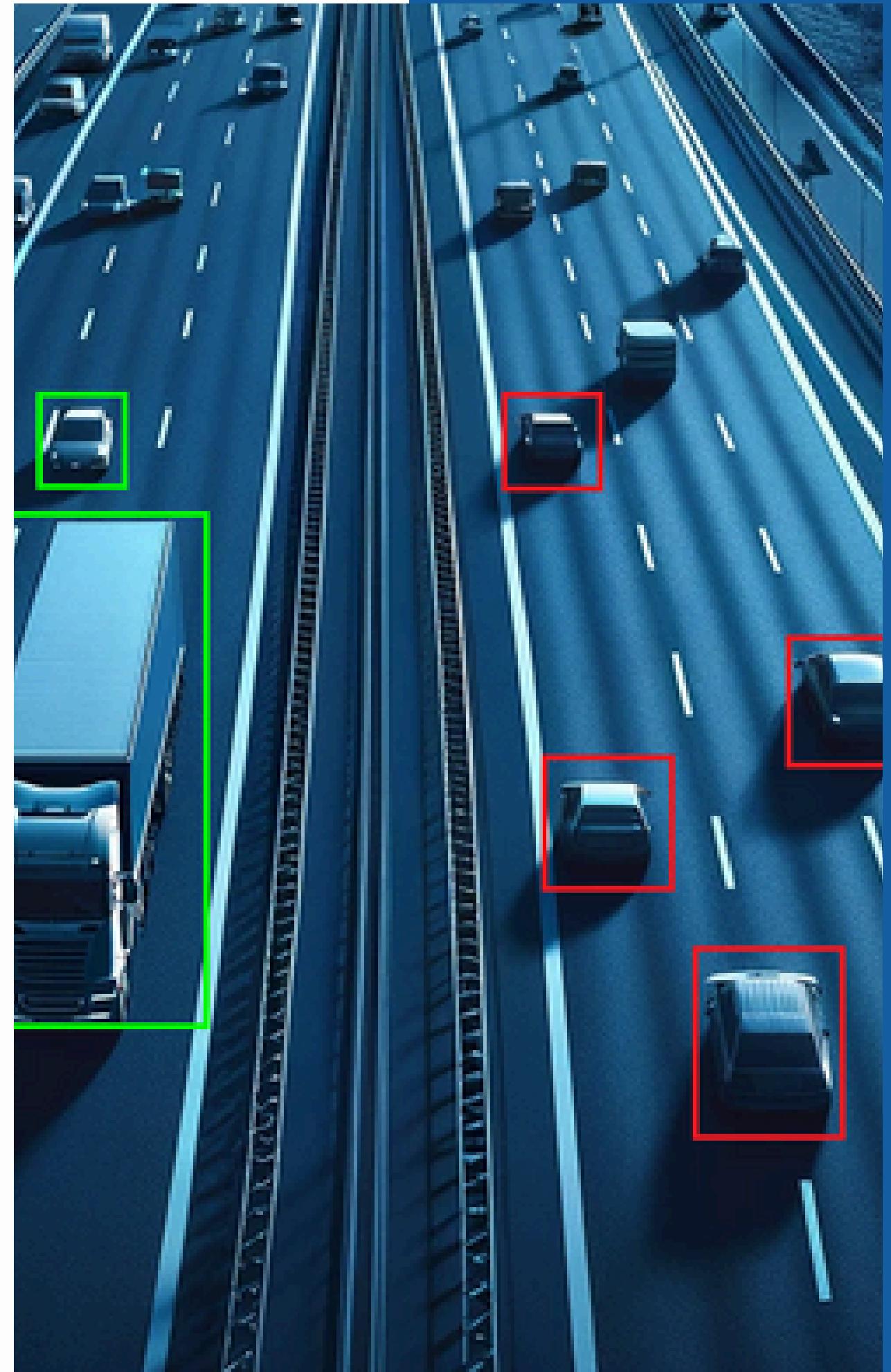
Ева Смилеска 221053

Стефан Ивановски 223257



Содржина

▶ Вовед	01
▶ Дефинирање на проблемот	02
▶ Архитектура на системот	03
▶ Алгоритми и технологии	04
▶ Имплементација на Video Processing	05
▶ Имплементација на Frontend и Backend интеракција	06
▶ Резултати и перформанси	07
▶ Добри страни и можности	08
▶ Позитивни исходи	09
▶ Заклучок	10





Вовед

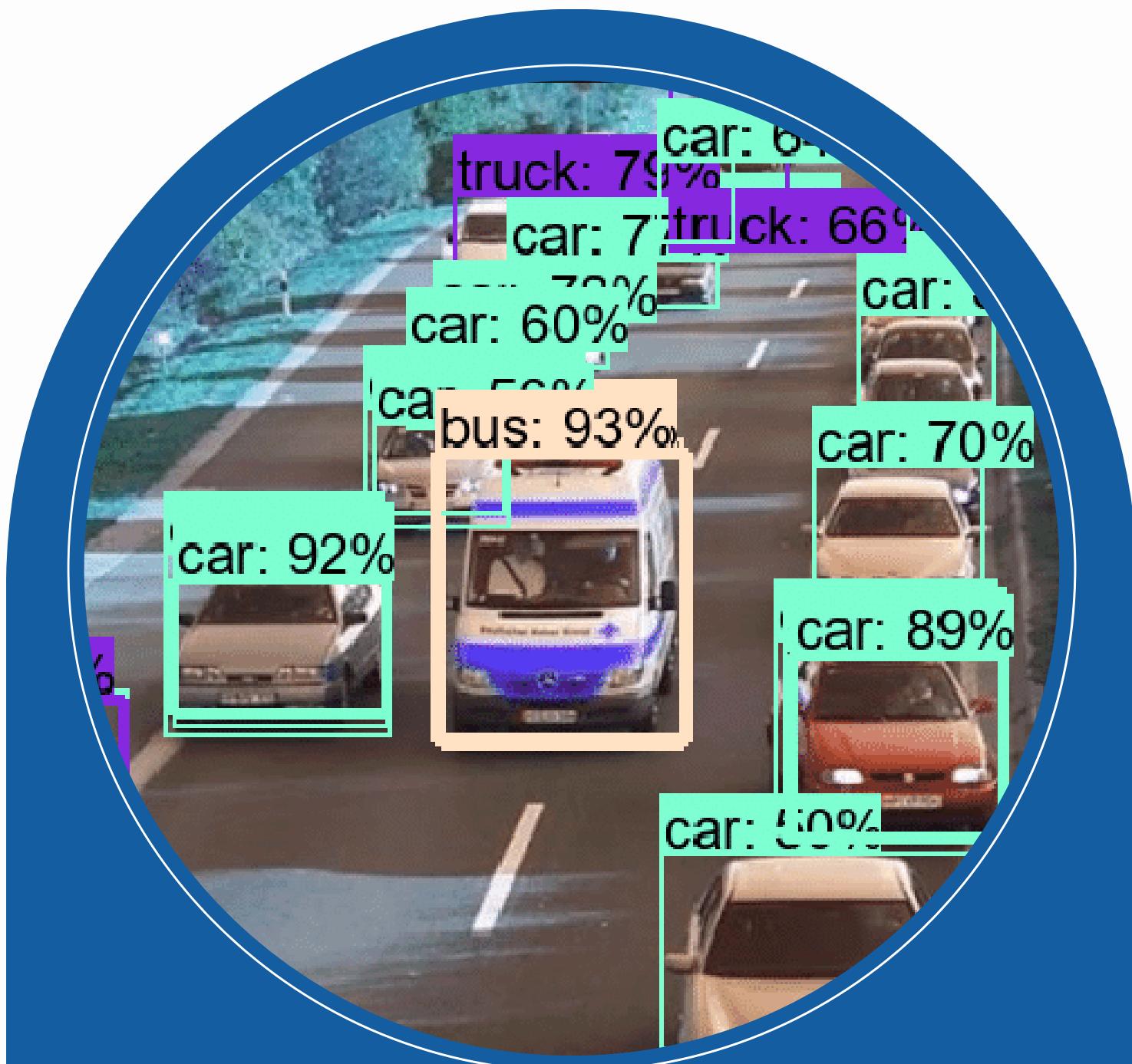
Овој проект има за цел да детектира и изброя возила во видео датотеки користејќи традиционални техники за обработка на слики. Фокусот е на примена на алгоритми за *background subtraction*, детекција на контури како и алгоритми за следење и бројење на возила што се движат во различни насоки. Системот е дизајниран да обработува претходно снимено видео, со иден потенцијал за следење во реално време во сообраќајните апликации.

Дефинирање на проблемот

Предизвици во детектирањето на возила

Откривањето возила во видео датотеките носи повеќе предизвици, како што се:

1. Справување со бучавата (noise) од сенките, промените на осветлувањето и рефлексите.
2. Обезбедување ефикасни перформанси за апликации во реално време.
3. Прецизно бројење и следење возила дури и во сценарија со густ сообраќај



Архитектура на системот

Преглед на Backend и Frontend

Backend

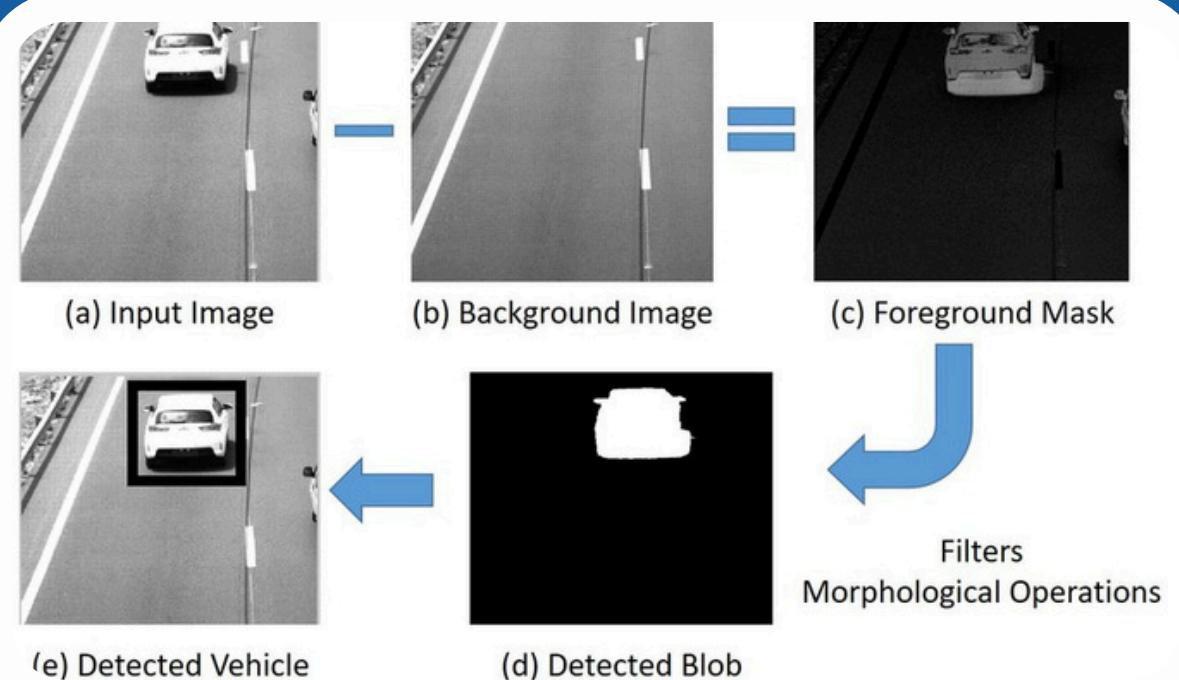
Изграден со користење на FastAPI за справување со прикачувања на видеа и OpenCV за детекција на возила.

Frontend

Изграден со користење на интерфејсот React.js за прикачување на видеа и прикажување на резултатите. Frontend-от е во интеракција со backend-от за да го прикаже бројот на возила по обработката.

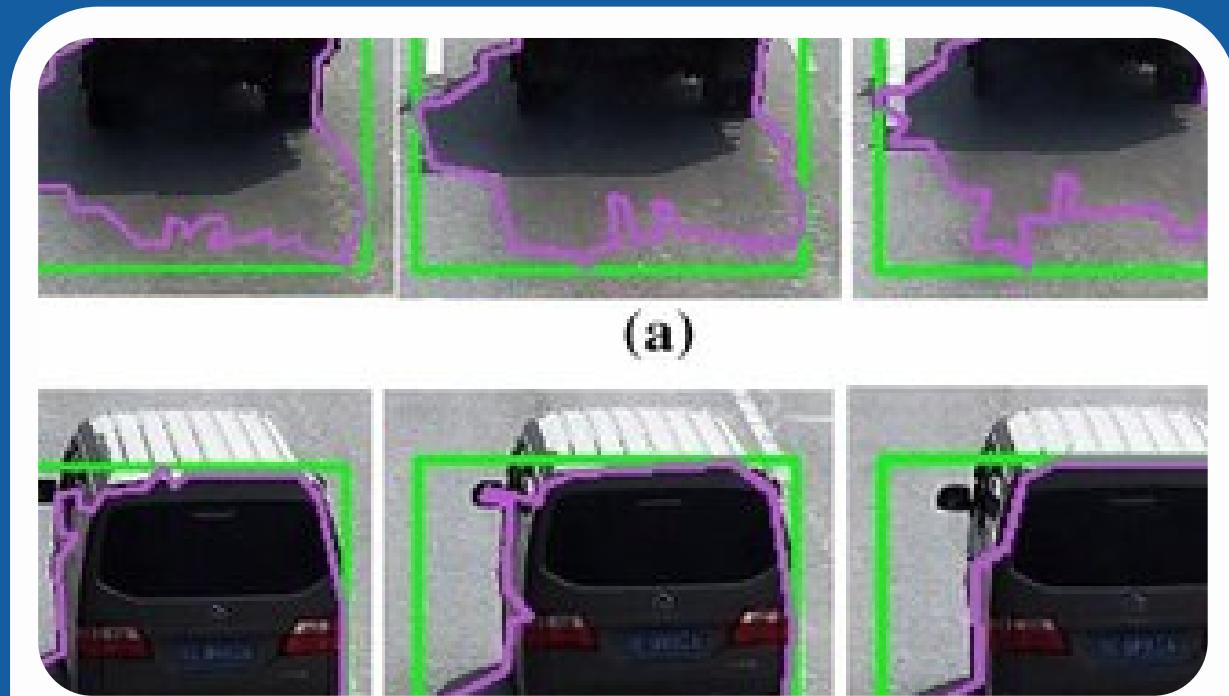


Алгоритми и технологии



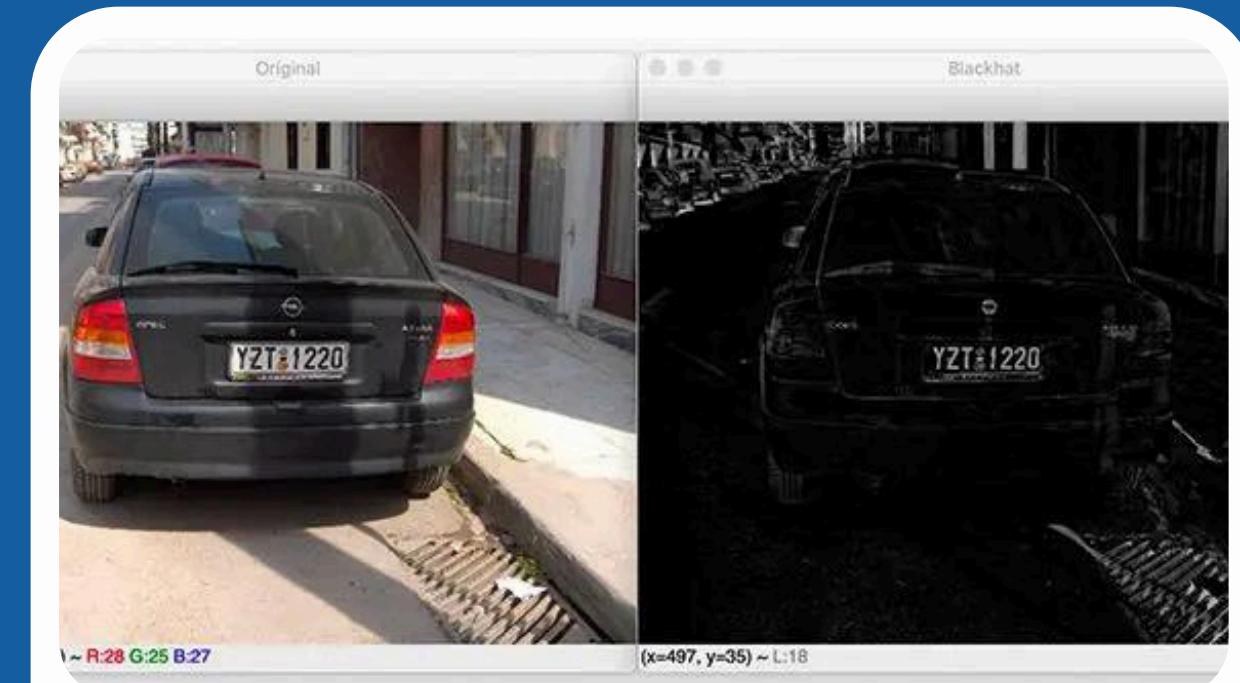
Background Subtraction (MOG2):

Одвоува возила во движење од статичната позадина во видео frames (рамки).



Откривање контура

Ги идентификува формите на возилото со анализа на контурите на објектите во обработеното видео.



Морфолошки операции:

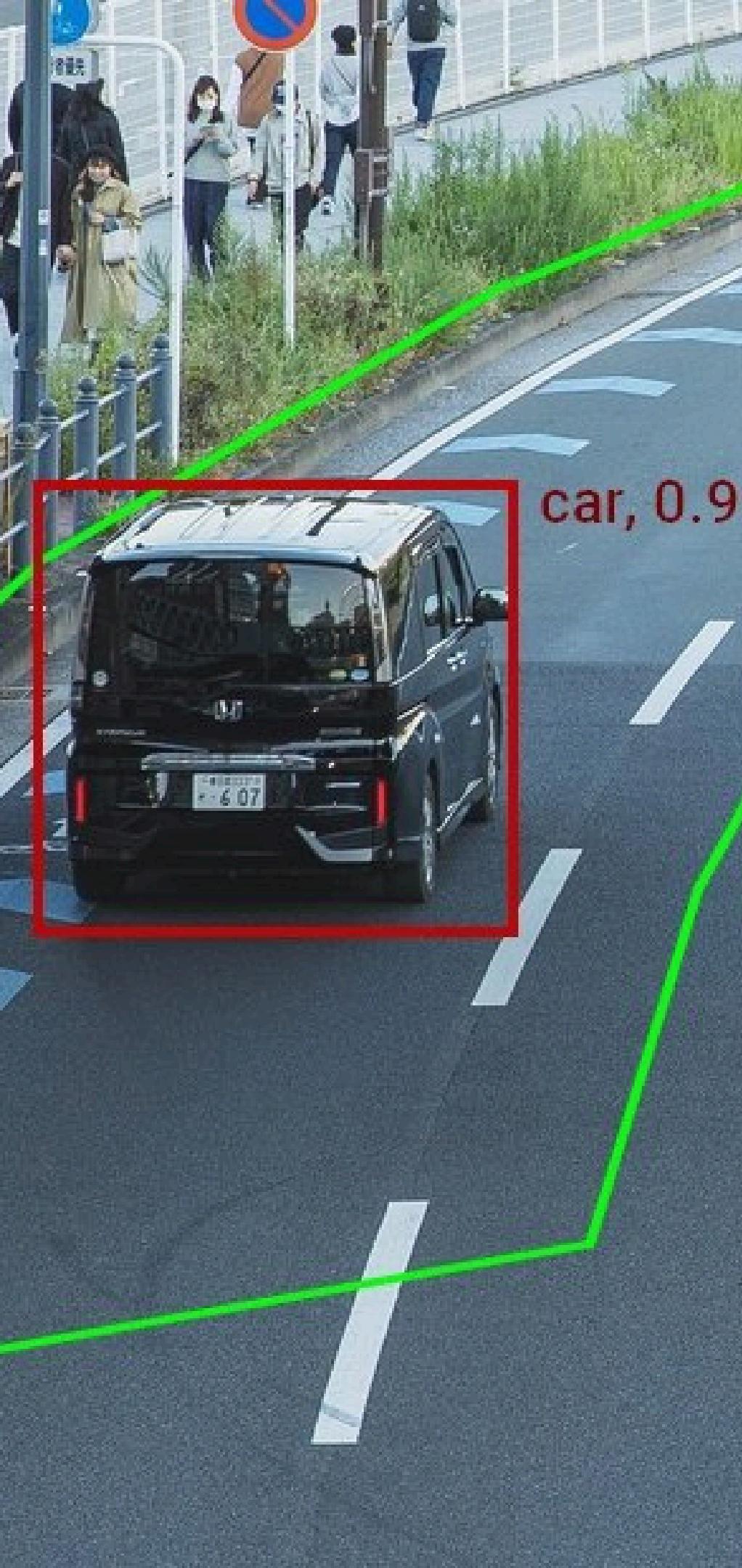
Ја подобрува точноста на откривање на возилото со чистење на бучавата (noise) од процесот на background subtraction.

Имплементација на Video Processing

Frame-by-Frame Processing

Секој frame (рамка) од видеото ги поминува следните чекори:

1. Конверзија на сиви тонови и замаглување за да се намали шумот (noise).
2. Примена на background subtractor за да се изолираат објектите што се движат.
3. Откривање на контури за да се идентификуваат валидни возила според нивната големина и форма.



Имплементација на Frontend и Backend интеракција

01

Frontend: Корисникот поставува видео преку React интерфејсот. Frontend-от го испраќа видеото до backend-от (FastAPI).

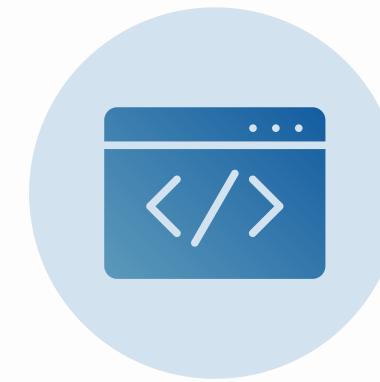
02

Backend: Го обработува видеото, применувајќи background subtraction и бројење возила.

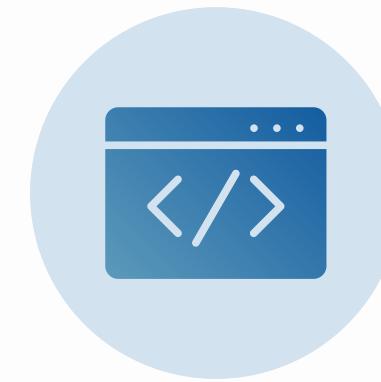
03

Резултат: Backend-от го враќа бројот на возила (кои се движат нагоре и надолу), прикажан во frontend-от.

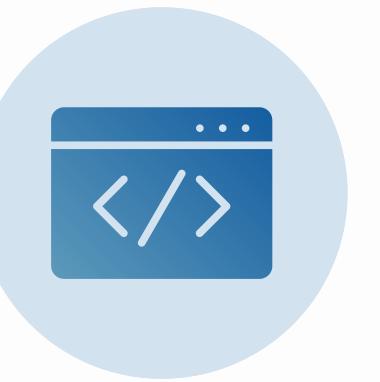
Резултати и перформанси



Системот успешно ги брои возилата во видеата, обезбедувајќи сигурни резултати во контролирани средини (controlled environments).



Перформансите се оптимизирани за претходно снимени видеа, а потребни се дополнителни подобрувања за обработка во реално време.



Примерите покажуваат точност во откривањето возила што се движат нагоре и надолу преку предодредена линија за откривање.

Добри страни и можности

Добри страни

1. Ефикасна обработка: Системот ефикасно се справува со претходно снимените видеа, обезбедувајќи брзи резултати со прецизна детекција на возилата.
2. Модуларен дизајн: Употребата на FastAPI и React.js обезбедува флексибилност и приспособливост, што го олеснува проширувањето или прилагодувањето на системот за различни случаи на употреба.
3. Ниска пресметковна цена: И покрај тоа што е изграден на традиционални техники за обработка на слики, системот функционира добро без потреба од длабоко учење, што го прави достапен за системи со помали хардверски барања.

Можности

Проширливост: Модуларната природа на системот овозможува лесна интеграција со дополнителни функционалности како обработка на видео во реално време, класификација или дополнителна оптимизација.





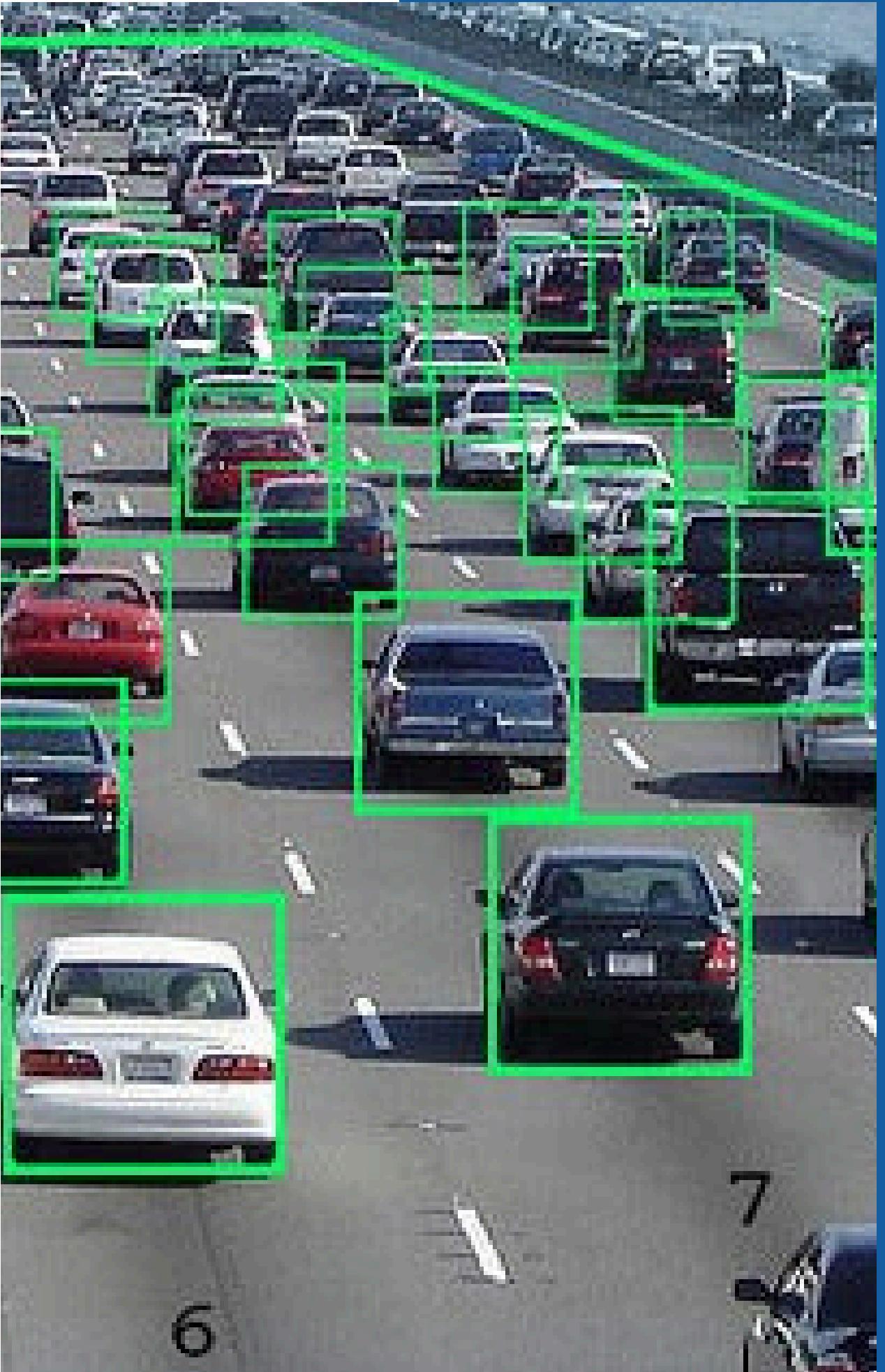
Позитивни исходи

Точна детекција на возила: Успешно имплементиран систем за детекција на возила кој ефикасно обработува видеа, обезбедувајќи точно пребројување на возила во различни услови.

Беспрекорна интеграција на backend: FastAPI служи како робустен backend, обезбедувајќи брзо време на одговор и лесно распоредување. Прилагодливост на различни случаи на употреба: Дизајнот на системот може лесно да се прилагоди на различни средини, без разлика дали се користи за следење на сообраќајот на паркинзи, автопати или дури и зони со слаб сообраќај.

Заклучок

Овој проект ја демонстрира ефикасноста на традиционалните техники за обработка на слики, како што се background subtraction и откривање на контури, во доменот на детекција на возила. Со модуларна архитектура, системот лесно се скалира за апликации во реално време. Идните подобрувања би можеле да вклучат модели за длабоко учење и анализа на видео со пренос во живо за посложени случаи на употреба.



**ВИ БЛАГОДАРИМЕ
ЗА ВНИМАНИЕТО!**

