



"Ss. Cyril and Methodius" University in Skopje
**FACULTY OF COMPUTER
SCIENCE AND ENGINEERING**

Проектна задача по предметот Вештачка интелигенција

Систем за препознавање возраст на детектирано лице со примена на машинско учење во обработка на слика

Професор:

д-р Андреа Кулаков

Изработиле:

Бојана Андонова 221225

Сандра Живановска 221152

Содржина

Вовед	3
1. Подготовка на податоци	4
1.1 Претпроцесирање на податоците.....	4
1.2 Поделба на податочното множество.....	5
2. Обучување на моделот	5
2.1 Што е VGG16 и зошто го користиме?	5
2.2 Тренирање, прилагодување и оптимизација на моделот	6
2.3 Евалуација на перформансите	11
3. YOLO и негова примена	12
4. Конечни резултати	13

Вовед

Развојот на машинското учење и машинската визија низ годините значително го има трансформирано пристапот кон обработка и анализа на визуелните податоци. Една од клучните аспекти во оваа област е анализата на лице, која вклучува различни задачи како што се идентификување на лица, детекција на емоции, одредување пол или проценка на возраст. Препознавањето на возраст претставува сложен проблем поради варијациите во изгледот на лицата и поради фактори како што се генетиката, начинот на живот и околината. Со помош на визуелните карактеристики на самото лице, како што се брчките, текстурата на кожата, обликот и други биолошки знаци кои се менуваат со текот на годините може приближно да се процени возраста.

Во рамките на оваа проектна задача, фокусот е да се развие модел за препознавање на возраст кој комбинира современи техники за детекција на лице во реално време и напредни методи за длабоко учење. Со цел да се постигне ова, ќе се користи YOLO (You Only Look Once) за детектирање на самото лице, кој е познат по својата брзина и точност, како и архитектурата VGG16 за тренирање на моделот. YOLO е алгоритам за детекција на објекти кој обработува дадена слика одеднаш, што го прави исклучително ефикасен за реално-временски апликации. Од друга страна, VGG16 е длабока невронска мрежа позната по својата способност да екстрахира детални и сложени карактеристики од слики, што ја прави идеална за задачи во кои е потребна анализа на истите.

Резултатите од ова истражување не само што ќе придонесат за подобрување на постоечките методи, туку и ќе отворат нови правци за истражување и примена во реални сценарија. На пример, овој модел може да се интегрира во интелигентни системи за надзор, адаптивни маркетинг стратегии или интерактивни AI-асистенти, каде што проценката на возраст е од клучно значење. Со оваа проектна задача, се стремиме кон создавање на робусен и ефикасен систем кој ќе биде корисен во различни практични апликации.

1. Подготовка на податоци

1.1 Претпроцесирање на податоците

За потребите на нашиот проект, го користиме UTK Face податочното множество, кое содржи над 20,000 слики на лица со означена возраст. Ова множество е широко користено во задачи поврзани со анализа на лице и препознавање на возраст, бидејќи нуди разновидни примероци со лица од различни етнички групи, возрасни групи и услови на осветлување. Секоја слика во множеството е именувана според формат кој ја означува возраста, полот и етничката припадност на лицето: **[возраст]_[пол]_[раса]_[датум и време].jpg**.

Оваа структура на именување ни овозможува лесно да ги извлечеме потребните ознаки за тренирање на моделот.

- **[возраст]**: Ова е цел број во опсег од 0 до 116, кој ја означува возраста на лицето на сликата. Оваа информација е клучна за нашата задача за препознавање на возраст.
- **[пол]**: Оваа вредност е или 0 - машко или 1 - женско, што ја означува половата припадност на лицето. Иако оваа информација не е директно поврзана со нашата задача, може да се користи за дополнителни анализи или подобрување на моделот.
- **[раса]**: Ова е цел број од 0 до 4, кој ја означува расната припадност на лицето. Вредностите се следни: 0 - Бела раса, 1 - Црна раса, 2 - Азијци, 3 - Индиска раса и 4 - Други. Иако оваа информација не е директно поврзана со препознавањето на возраст, ја користиме за истражување на влијанието на расната припадност врз точноста на моделот.
- **[датум и време]**: Ова е временска ознака во формат ууууmmddHHMMSSFFF, која го означува датумот и времето кога сликата е собрана во UTKFace множеството.

За да ги подготвиме сликите за тренирање на моделот, секоја слика се обработува преку серија чекори за да се осигура дека е во соодветен формат. Прво, сликата се вчитува од дадената патека и се декодира од JPEG формат во тензор, кој претставува матрица од пиксели со три канали (црвено, зелено и сино). Потоа, сликата се менува на фиксна големина, односно 224 x 224 пиксели, што е стандарден влезен формат за многу модели за длабоко учење, вклучувајќи го и VGG16. Следно, вредностите на пикселите, кои првично се во опсегот [0, 255], се нормализираат во опсегот [0, 1] со делење на секоја вредност со 255. Оваа нормализација е важна за да се осигура дека моделот може ефикасно да ги обработи податоците и да постигне подобри резултати. На крај, претпроцесираната слика се враќа како тензор, подготвен да биде внесен во невронската мрежа. Овој процес е клучен за подготовка на податоците и за да осигура конзистентност во влезните податоци, што е неопходно за успешно тренирање на моделот.

1.2 Поделба на податочното множество

За да се осигура дека моделот ќе може да генерализира добро на нови податоци, податочното множество е поделено на три дела: тренирачко, валидациско и тест множество. Оваа поделба е извршена со случаен метод, но со фиксна почетна состојба (`random_state=42`) за да се осигура повторливост на резултатите. Со други зборови, иако податоците се делат случајно, поставувањето на почетна состојба гарантира дека поделбата ќе биде иста секој пат кога ќе се изврши кодот. Најголемиот дел од податоците т.е. 80% се користат за тренирање на моделот. Останатите податоци се дополнително поделени на два дела: еден дел или 10% се користи за валидација за време на тренирањето, а другиот дел, исто така 10%, се задржува како тест множество за конечна евалуација. Валидациското множество служи за следење на перформансите на моделот за време на тренирањето и за прилагодување на хиперпараметрите, додека тест множеството, кое е целосно независно, се користи за да се процени колку добро моделот функционира на невидени податоци. Овој пристап е клучен за спречување на `overfitting` и за осигурање дека моделот е способен да прави точни предвидувања во реални сценарија.

2. Обучување на моделот

Во овој дел, ќе го опишеме целиот процес на истражување и тренирање на моделот за препознавање на возраст. Ова вклучува избор на соодветна архитектура, прилагодување на моделот за нашата специфична задача, процесот на тренирање и евалуација на перформансите. Почнувајќи од користењето на VGG16 како основа, ќе објасниме како оваа архитектура е прилагодена за нашата задача на препознавање на возраст, како се извршува тренирањето и како се избира најоптималниот модел врз основа на неговата точност и способност за генерализација. Дополнително, ќе ги анализираме загубите (`loss`) на тренирачкото и валидациското множество за да го оцениме однесувањето на моделот и да го спречиме претренирањето.

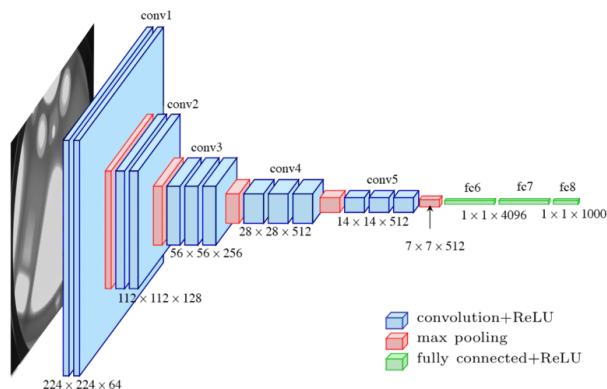
2.1 Што е VGG16 и зошто го користиме?

VGG16 е една од најпознатите архитектури за конволуциски невронски мрежи, развиена од Visual Geometry Group - VGG на Универзитетот во Оксфорд. Овој модел е познат по својата едноставна, но моќна структура, која се состои од 16 слоеви, вклучувајќи 13 конволуциски слоеви и 3 целосно поврзани или `fully connected` слоеви. Конволуциските слоеви се одговорни за екстракција на карактеристики од сликите, додека целосно поврзаните слоеви ги користат овие карактеристики за класификација. VGG16 е обучен на

милиони слики од ImageNet базата на податоци, што го прави исклучително ефикасен во препознавањето на визуелни шаблони.

Во нашиот проект, VGG16 е користен како базен модел, а потоа истиот е прилагоден (fine-tuned) за нашата специфична задача со замрзнување на повеќето негови слоеви и отмрзнување на последните неколку слоеви за да може да се прилагоди на нашите податоци. Ова е направено за да се искористи знаењето што моделот веќе го има од претходното тренирање на ImageNet, но истовремено да се овозможи учење на специфични карактеристики поврзани со возраста на лицата. На врвот на VGG16 архитектурата, додаваме дополнителни слоеви, кои помагаат во подобрување на перформансите и спречување на претренираност - overfitting.

VGG16 е адекватен поради неговата докажана ефикасност во задачи поврзани со анализа на слики, како и поради неговата способност да екстрахира детални и сложени карактеристики од сликите. Ова е особено важно за препознавањето на возраст, каде што мали детали како брчки, текстура на кожата и облик на лицето играат клучна улога. Со користење на VGG16 како основа, нашиот модел може да постигне висока точност и робустност во предвидувањето на возраста.



Слика 1 Архитектура на VGG16

2.2 Тренирање, прилагодување и оптимизација на моделот

Во првичната фаза на тренирање, моделот е изграден врз основа на VGG16 архитектурата, каде што **сите слоеви на VGG16 се замрзнати освен последниот целосно поврзан слој**. Ова значи дека моделот го користи знаењето стекнато од претходното тренирање на ImageNet, но дозволува само мали прилагодувања на последниот слој за да се прилагоди на нашата специфична задача за препознавање на возраст. На врвот на VGG16 архитектурата, додаваме дополнителни слоеви:

1. **Flatten слој** – За да се претвори излезот од VGG16 во едно-димензионален вектор.
2. **Dense слој со 256 неврони и ReLU активација** – За да се овозможи моделот да научи посложени карактеристики.

3. **Dropout слој со стапка 0.3** – За да се спречи overfitting со случајно исклучување на 30% од невроните за време на тренирањето.
4. **Dense слој со 128 неврони и ReLU активација** – Дополнителен слој за подобрување на капацитетот за учење.
5. **Dropout слој со стапка 0.3** – Повторно исклучување на 30% од невроните за дополнителна регуларизација.
6. **Излезен Dense слој со 1 неврон и линеарна активација** – За регресија на возраста, односно за предвидување на бројот на години на детектираното лице.

Моделот е компајлиран со **Adam оптимизатор** и **рата на учење од 0.01**, како функција на загуба беше користена **Mean Squared Error (MSE)**, а како метрика за евалуација беше користена **Mean Absolute Error (MAE)**. Тренирањето е извршено во 40 епохи.

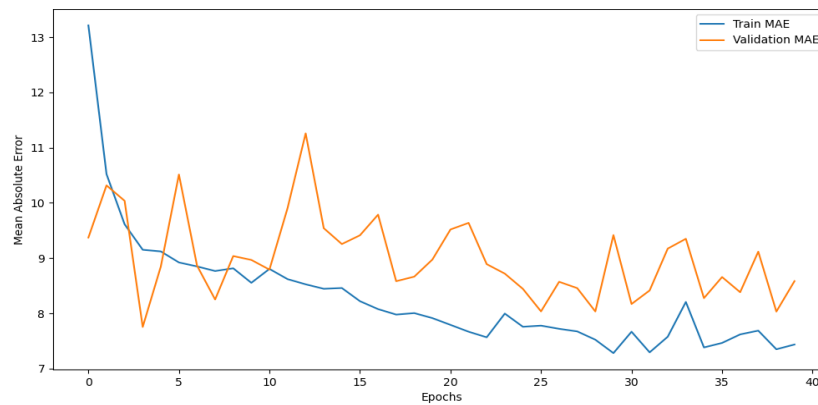


График 1 Фаза 1 од тренирањето

Од графикот на загубите прикажан на Слика 2, може да се забележи дека моделот покажува знаци на **overfitting**. Иако загубата на тренирачкото множество (Train MAE) константно се намалува, валидациската загуба (Validation MAE) почнува да се зголемува по одреден број епохи. Ова укажува на тоа дека моделот премногу се прилагодува на тренирачкото множество и губи способност да генерализира добро на нови податоци. **За да се спречи ова, одлучивме да го зголемиме Dropout слојот од 0.3 на 0.5**, со цел да се намали претренираноста и да се осигура подобар баланс меѓу учењето и генерализацијата.

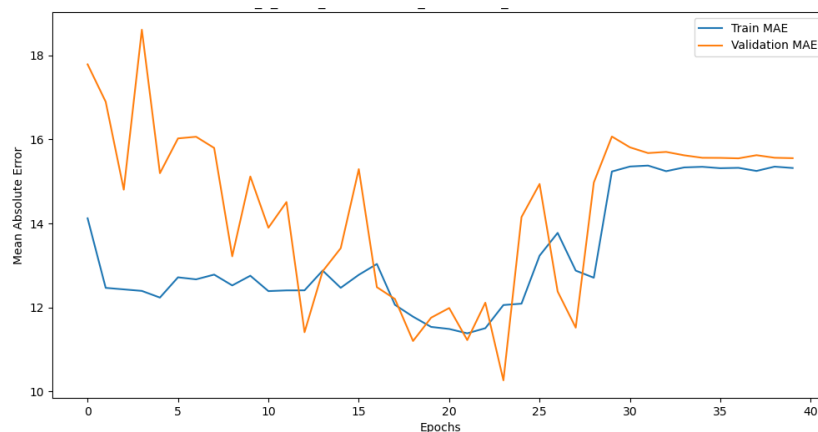


График 2 Фаза 2 од тренирањето

Сепак, на Слика 3 се забележуваат големи осцилации, што е индикација за нестабилно тренирање. Ова најверојатно се должи на превисок learning rate (во случајот 0.01), кој предизвикува тежините на моделот да се ажурираат со преголеми чекори, што резултира со „скокање“ наместо постепена оптимизација. За да се реши овој проблем, во следниот модел ја **намалуваме ратата на учење од 0.01 на 0.0001**.

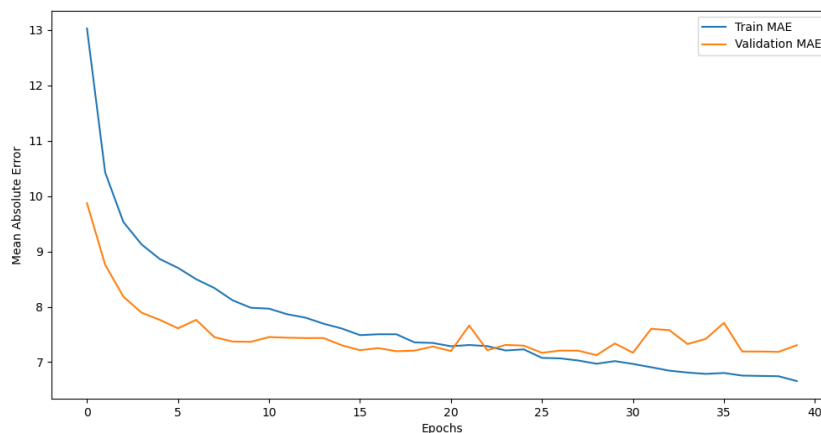


График 3 Фаза 3 од тренирањето

Во продолжение на анализата, од Слика 4 увидуваме дека осцилациите не се толку изразени, но сепак постои простор за подобрување на перформансите на моделот. Како следен чекор, пробуваме да воведеме **Leaky ReLU** активациска функција наместо стандардната ReLU. Leaky ReLU е дизајнирана да го реши проблемот на „мртвите неврони“, кој може да се јави кај ReLU, затоа што невроните не се активираат за негативни влезови. Оваа функција дозволува мали, ненулни негативни вредности кога влезот е негативен, што може да го подобри капацитетот на моделот да научи посложени карактеристики. Со воведување на Leaky ReLU, моделот може подобро да ги искористи сите неврони во секој слој, што би можело да доведе до подобри резултати и на тренирачкото и на валидациското множество.

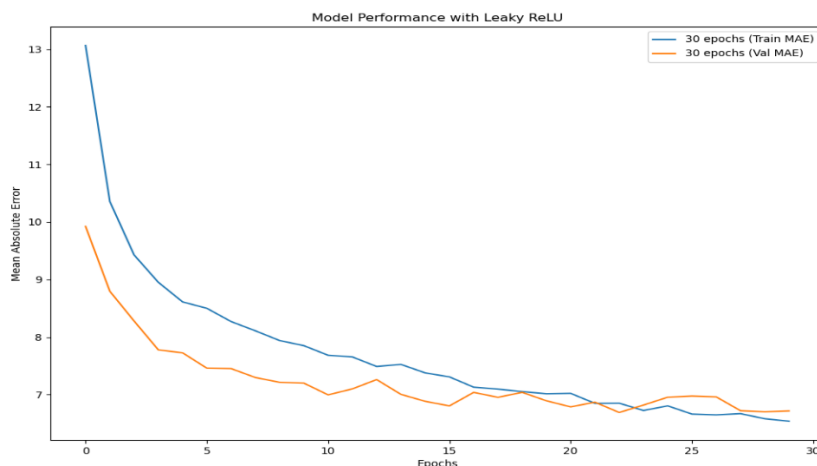


График 4 Фаза 4 од тренирањето

Воведувањето на Leaky ReLU, како што гледаме на Слика 5, не доведе до значително подобрување на перформансите на моделот, што укажува на тоа дека проблемот не е решен само со промена на активациската функција. Затоа, како следен чекор, одлучуваме да ги одмрзнеме последните четири слоеви од VGG16 архитектурата и да додадеме дополнителен Dense (целосно поврзан) слој со 512 неврони, пред другите слоеви, но над базниот VGG модел.

Конкретно, ги **одмрзнуваме последните три целосно поврзани (fully connected) слоеви** и еден конволуциски слој. Со овие слоеви, моделот добива поголема флексибилност да ги прилагоди карактеристиките кон нашиот проблем, што може да доведе до подобрување на перформансите, додека додавањето на нов слој со повеќе неврони му овозможува на моделот да научи уште посложени односи, да го зголеми својот капацитет за репрезентација и да ги извлече специфичните карактеристики од сликите.

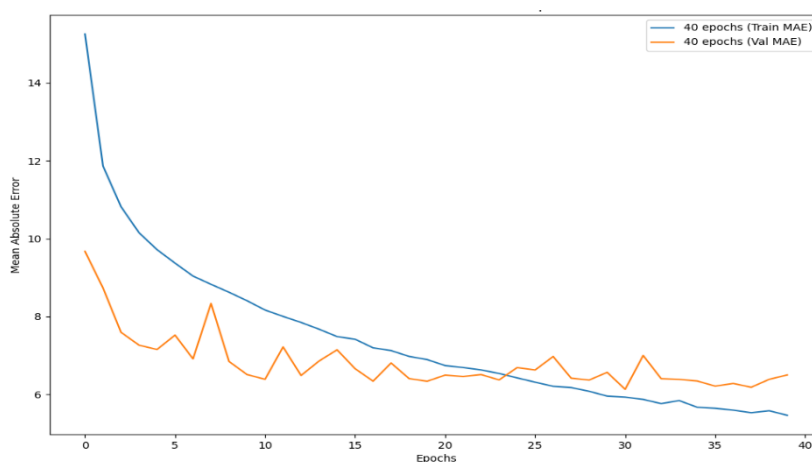


График 5 Фаза 5 од тренирањето - додаден нов слој од 512 неврони

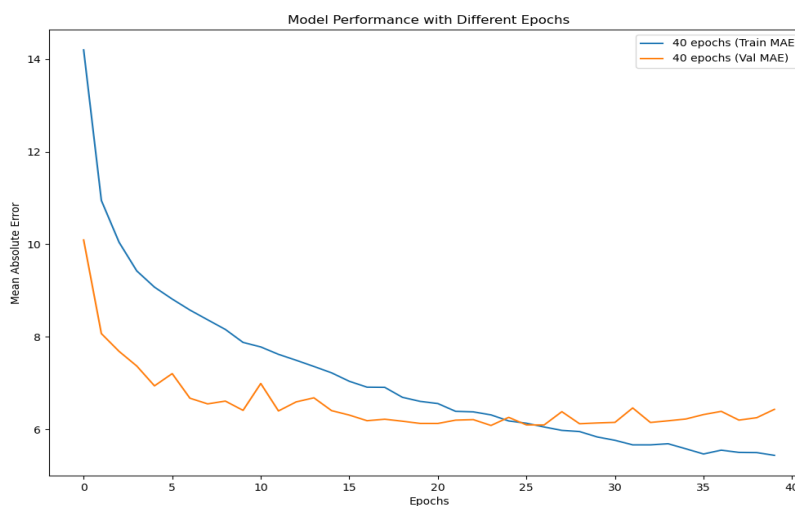


График 6 Фаза 5 од тренирањето - БЕЗ додаден нов слој од 512 неврони

Во продолжение, се обидуваме да го подобриме моделот со **одмрзнување на дополнителни 2 слоја, односно вкупно 6 од последните слоеви** од VGG16 архитектурата. Идејата е дека со одмрзнување на повеќе слоеви, моделот ќе може да научи уште посложени и поспецифични карактеристики кои може да бидат клучни за подобрување на перформансите. Сепак, од Слика 8, може да се воочи дека моделот со 6 одмрзнати слоеви не покажува значително подобрување во споредба со моделот каде што беа одмрзнати само 4 слоја – Слика 7. Резултатите се слични и за други метрики како R2 score. Одмрзнувањето на премногу слоеви безпотребно ја зголемува комплексноста на моделот и доведе до малку поголема нестабилност во загубите, односно не успеа да постигне значително подобрување. Затоа, **моделот со 4 одмрзнати слоја е подобар избор**. Поедноставен е и покажува за нијанса подобар баланс меѓу учењето и генерализацијата.

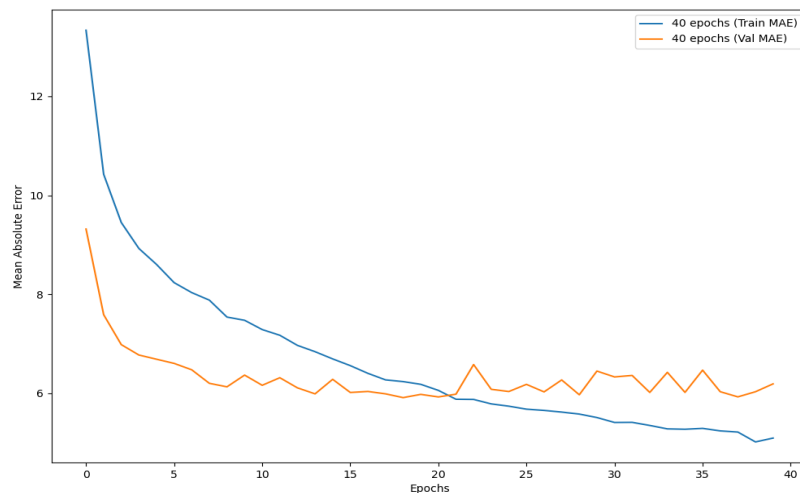


График 7 Фаза 6 од тренирањето - одмрзнати 6 слоја

Поради знак на прекумерно прилагодување или overfit после 20-тата епоха го намалуваме бројот на епохи во тренирањето. Со ова заклучуваме дека оптимален избор на модел е:

- основа VGG16 со одмрзнати 4 слоја
- Flatten слој
- Dense слој со 256 неврони и ReLU активација
- Dropout слој со стапка 0.3
- Dense слој со 128 неврони и ReLU активација
- Dropout слој со стапка 0.3
- Излезен Dense слој со 1 неврон и линеарна активација
- Adam оптимизатор
- Рата на учење 0.0001

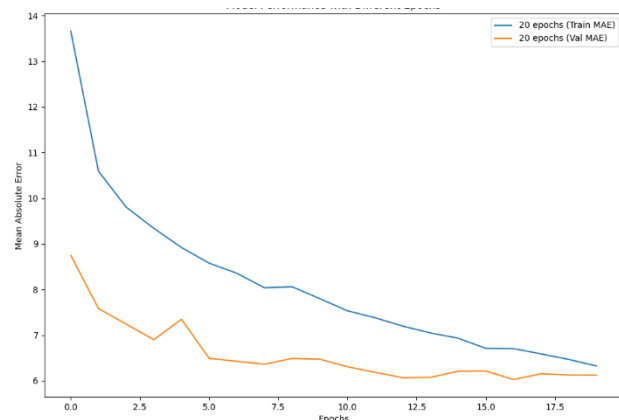


График 8 Конечен резултат

2.3 Евалуација на перформансите

Анализата на перформансите на оптималниот модел, врз основа на возрасните и етничките групи, овозможува да се идентификуваат групите каде моделот најмногу греша, што може да укаже на потенцијални слабости или пристрасности во моделот. Оваа анализа ќе вклучува преглед на R^2 скорот за различни групи, како и проценка на Mean Absolute Error (MAE) за секоја возрасна и етничка категорија од тест множеството.



График 9



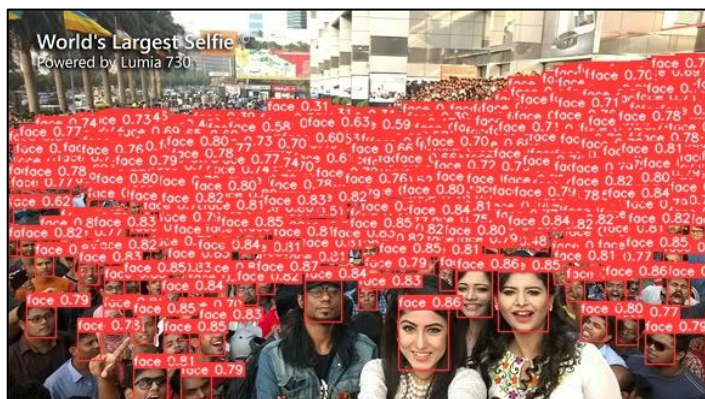
График 10

R^2 Score: 0.79389488697052

MAE: 6.390516757965088

3. YOLO и негова примена

YOLO (You Only Look Once) е современ алгоритам за детекција на објекти кој за разлика од традиционалните методи, кои користат повеќе чекори, обработува цела слика наеднаш. Овој алгоритам се состои од 24 конволуциски слоеви, четири max-pooling слоеви и два целосно поврзани слоеви. Првиот чекор во YOLO е поделба на оригиналната слика во $N \times N$ матрица



Слика 2 Детекција на лица, користејќи YOLO

од еднакви квадрати. Секој квадрат е одговорен за локализирање и предвидување на класата на објектот кој го покрива. Следниот чекор е одредување на рамките (bounding boxes) кои ги означуваат сите објекти во сликата. Во една слика може да има многу рамки или кутии, колку што има објекти. Најчесто, еден објект може да има повеќе кандидати за предвидување од различни квадрати, но не сите се релевантни. За таа цел се пресметува вредност на доверба (confidence value). Оваа вредност претставува мерка на сигурност со која моделот верува дека одреден објект е присутен или не. Станува збор за број помеѓу 0 и 1, каде што 1 значи целосна доверба дека објектот е точен и прецизно локализиран, додека 0 значи дека моделот не верува дека во таа рамка има објект.

YOLO е познат по својата способност да ги балансира брзината и точноста, што го прави еден од најпопуларните алгоритми за детекција на објекти во областа на машинската визија. Во нашиот случај, со користење на YOLO, можеме брзо и прецизно да ги детектираме лицата во реално време, односно директно на вклучена камера. Ова е основниот чекор пред да се примени моделот за предвидување на возраст. Ја користиме класата 0 бидејќи таа одговара на детекцијата на лица, што е нашата примарна цел. Потоа ја поставуваме вредноста на доверба на 0.5 за да ги филтрираме само оние предвидувања кои имаат доволно висока сигурност за присуство на објект или лице. Оваа праг вредност ни овозможува да ги елиминираме непрецизните и да ги задржиме само релевантните детекции.

Со ова, YOLO ни овозможува да ги изолираме регионите од сликата каде што се наоѓаат лицата, што значително ја намалува сложеноста на податоците и го подобрува квалитетот на предвидувањата. Ова е особено важно за нашиот модел каде што точната локација на лицето е клучна за последователната анализа. Овој алгоритам ни овозможува да создадеме ефикасен и скалабилен систем кој може да се користи во различни реални сценарија, од безбедносни камери до персонализирани маркетинг платформи. Со YOLO како наша основна алатка за детекција, осигуравме дека нашиот модел за предвидување на возраст е не само точен, туку и практичен за реална употреба.

4. Конечни резултати

