Универзитет „Св. Кирил и Методиј“
Факултет за информатички науки и компјутерско инженерство
д-р Ивица Димитровски

Детекција на лица и замена на лице

Ангела Ангелеска, 221083
Елена Талеска, 223141

Содржина

1. Апстракт

Оваа семинарска работа проучува алгоритми за детекција и замена на лица, со цел да се илустрираат техничките аспекти и практичната имплементација во Python. Прво, ќе бидат презентирани неколку алгоритми за детекција на лица, со објаснување на нивните механизми и применливост. Потоа, ќе се разгледа алгоритам за замена на лица, кој овозможува создавање на нови визуелни прикази. Завршно, во семинарската работа ќе се презентира конкретна имплементација на овие алгоритми во Python, со демонстрација на кодот и анализа на постигнатите резултати.

2. Вовед

Во денешната ера на брз напредок на технологијата, дигиталното процесирање на слика игра клучна улога во нашето секојдневие. Меѓу различните области на оваа сфера, детекцијата и замената на лица се особено значајни, со широк спектар на примени во различни индустрии.

Детекцијата на лица се однесува на процесот на идентификација и анализа на човечки лива во слики или видеа. Оваа техника е основа за бројни модерни апликации, вклучувајќи системи за препознавање на лица, видеонадзор и безбедносни технологии.

Од друга страна, замената на лица воведува иновативни методи кои овозможуваат трансформирање на едно лице во друго, создавајќи визуелни ефекти што се користат во филмови, видеоигри и социјални медиуми. Со развојот на напредни алгоритми и техники, овие технологии стануваат сè попрецизни и подостапни за користење. Напредокот во дигиталните ресурси и квалитетот на камери дополнително ја олеснуваат детекцијата и замената на лица, отворајќи нови можности за креативност и иновации во визуелната уметност и медиумите.

3. Алгоритми за детекција на лица

3.1 Dlib Frontal Face Detector

Детекторот на лица на dlib се базира на алгоритам за машинско учење наречен Histogram of Oriented Gradients (HOG), комбиниран со линеарен класификатор.

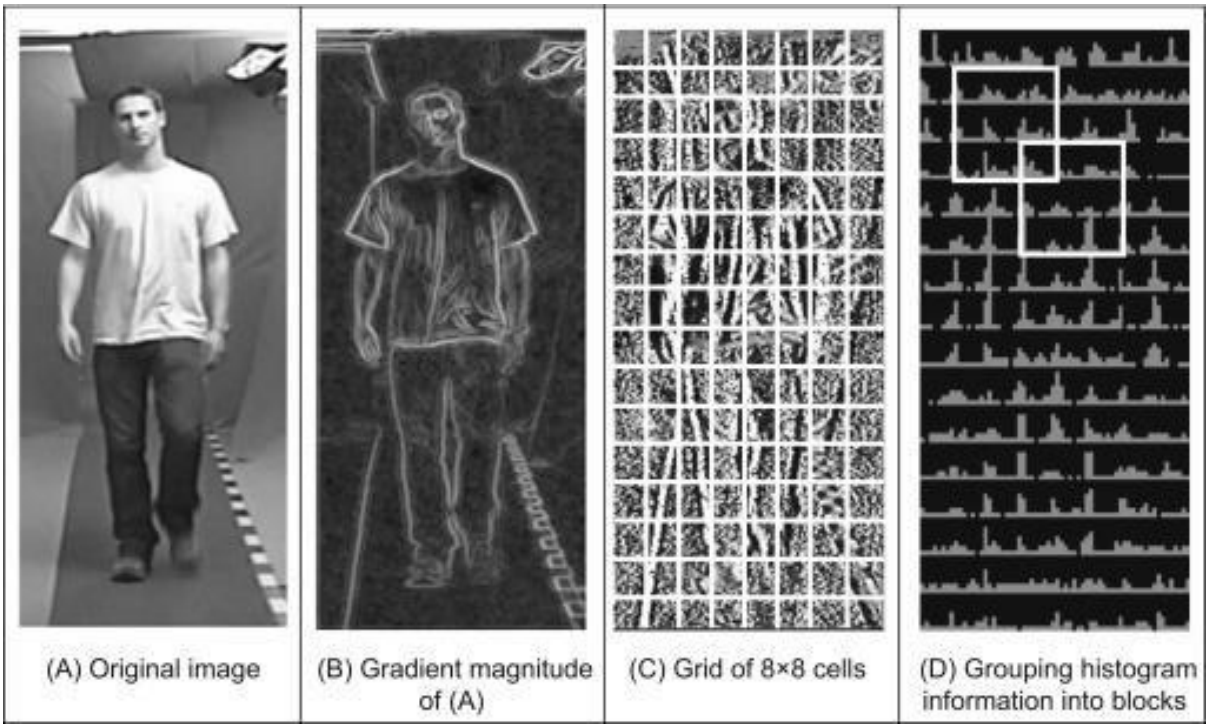
HOG е техника за препознавање на објекти која се фокусира на анализа на градиентите (промени во интензитетот на пикселите) на сликата. Процесот се состои од следните чекори:

- Пресметување на градиенти: Градиентите се пресметуваат за секој пиксел во сликата и укажуваат на присуство на рабови или текстури.
- Делење на сликата: Сликата се дели на мали ќелии (обично 8x8 или 16x16 пиксели). За секоја ќелија се создава хистограм на ориентираните градиенти.
- Нормализација: Хистограмите на градиентите во секоја ќелија се нормализираат, со цел осветлувањето и контрастот да не влијаат на препознавањето. Ова се постигнува со групирање на ќелиите во блокови (обично 2x2 ќелии).
- Формирање на вектор на карактеристики: Нормализираните хистограми од сите блокови се комбинираат во долг вектор, кој го претставува изгледот на објектот во сликата. Овој вектор служи за идентификација на објекти, вклучувајќи и лица.

Линеарен класификатор е алгоритам кој применува математички формули за донесување одлуки. Тој определува дали карактеристиките на сликата (на пример векторот на карактеристики на HOG) припаѓаат на категоријата „лице“ или не.

Класификаторот е трениран на голем број на слики кои се означени како слики на кои има лица или не. Со овие примери, тој учи да ги идентификува

клучните карактеристики кои ги разликуваат лицата од другите објекти. Откако ќе биде истрениран, класификаторот користи математички формули за да пресмета дали новата слика содржи лице. Тој го прави ова така што ги проверува карактеристиките на сликата и ги споредува шаблоните што ги научил.



Слика 1. Histogram of Oriented Gradients (HOG)

3.2 Eigenfaces

Eigenfaces е алгоритам за препознавање на лица базиран на метода наречена Principal Component Analysis (PCA). PCA е техника што се користи за намалување на димензионалноста на податоците и идентификување на најважните карактеристики.

3.3 Fisherfaces

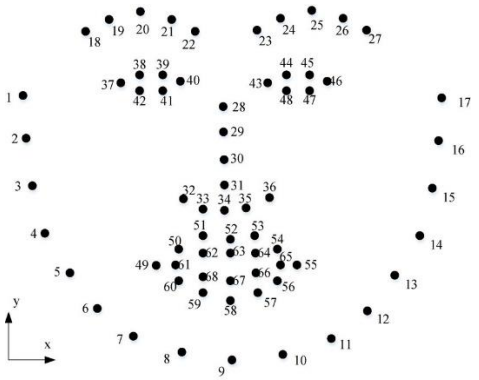
Fisherrfaces е алгоритам што се базира на Linear Discriminan Analysis (LDA), а се користи за препознавање на лица. LDA е техника што се користи за оптимизирање на разликувањето меѓу различни класи во податоците.

4. Алгоритам за замена на лица

Во оваа семинарска работа, разгледуваме и алгоритам за замена на лица кој комбинира неколку познати техники од областа на компјутерската визија. Алгоритамот се состои од неколку фази, од откривање на лица до клонирање на лицето.

Првата фаза е детекција на лицето, за што се користи алгоритамот Histogram of Oriented Gradients (HOG). HOG го анализира интензитетот на светлината на сликата преку ориентација на градиентите и ја дели сликата на помали сегменти наречени „ќелии“. Овој метод е робустен на промени во осветлувањето и помага прецизно да се детектираат лицата и нивните карактеристики.

Откако ќе се детектира лицето, алгоритамот користи shape predictor од библиотеката dlib за да идентификација на 68 клучни точки на лицето, како веѓи, очи, нос и уста. Овие точки се од витално значење за понатамошните фази на трансформација, бидејќи служат како водич за геометриските промени на лицето.



Слика 2. Фацијални ознаки

Следниот чекор е примената на Делаунаова триангулација. Овој метод го дели лицето на мали триаголници. Секој триаголник може да се деформира за да едното лице се адаптира на обликот на другото лице.

За трансформација на триаголниците добиени со Делаунаовата триангулација, користиме Афинска трансформација. Таа овозможува геометриско искривување на триаголниците од изворното лице така што ќе се совпаднат со триаголниците на целното лице.

Последниот чекор е беспрекорно клонирање на лицата, каде се користи техниката Poisson blending. Оваа техника овозможува минимизирање на видливоста на рабовите помеѓу деловите на лицето и создава мазен и природен преод меѓу изворното и целното лице. На овој начин, резултантната слика изгледа природно и реалистично, без видливи артефакти.

5. Имплементација во Python

Во овој дел од семинарската работа, ќе го опишеме кодот за имплементација на алгоритамот за детекција и замена на лица. Проектот се потпира на библиотеки како што се OpenCV, dlib и Numpy, како и Tkinter за интеракција со корисникот. Овие алатки овозможуваат прецизно обработување на слики и детектирање на лица, што е клучно за успешна реализација на задачата.

Процесот започнува со вчитување на две слики, при што корисникот избира фотографии преку графички интерфејс создаден со Tkinter. Првата слика е изворното лице, а втората целното лице.

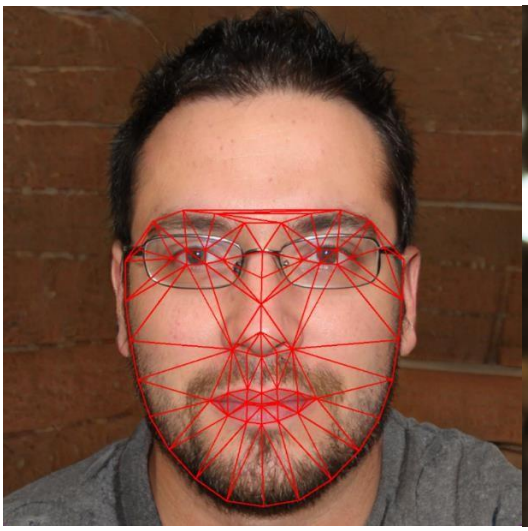
Слика 3. Изворно лице

Слика 4. Целно лице

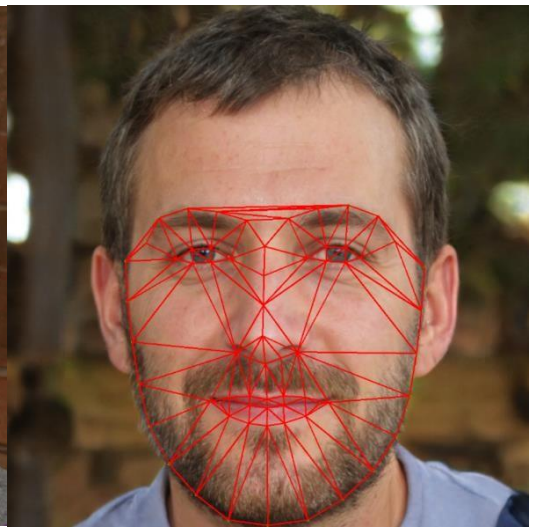
Следниот чекор е детекцијата на лица, за што се користи dlib. За подобра обработка, сликата се конвертира во сива скала, а потоа се применува детекторот на лица. Детекторот на лица од dlib враќа координати на регионот од интерес за секое лице на сликата. За дополнителна прецизност, се користи dlib.shape_predictor за идентификација на 68 клучни точки на лицето, како што се очите, носот и устата.

Слика 5. Значајни точки на лицето

Откако клучните точки се добиени, следува Delaunay триангулација. Оваа техника создава мрежа од триаголници врз основа на клучните точки на лицето. Прво, се создава конвексна хул со cv2.convexHull, а потоа се користи cv2.Subdiv2D за да се направи Delaunay триангулација. Индексите на триаголниците се чуваат за да се користат во следните чекори на обработка.



Слика 6. Триангулација на изворното лице



Слика 7. Триангулација на целното лице

За секој триаголник од изворното лице, се прави афинска трансформација кон соодветниот триаголник од целното лице. Овој процес вклучува извлекување на пиксели од оригиналното лице со помош на маски, што овозможува прецизно комбинирање на деловите на лицата.

Откако сите триаголници се трансформираат, се комбинираат во новата слика. За комбинирање на новите триаголници со остатокот на целното лице се користи функцијата cv2.add, која ги обединува различните делови. На крајот, за да се постигне природен изглед на заменетото лице, се применува cv2.seamlessClone, што овозможува мазен премин на текстурите и тоновите на лицата.

На крајот, се прикажува финалната слика преку cv2.imshow.

Слика 8. Резултат

6. Заклучок

Во оваа семинарската работа, го разгледавме процесот н детекција и замена на лица со помош на современи алгоритми и библиотеки како што се OpenCV и dlib. Применивме различни техники за детекција на лице, како што се Dlib Frontal Face Detector, кој користи Histogram of Oriented Gradients (HOG) и линеарен класификатор за точна идентификација на лицата во сликите. Овој детектор покажува висока прецизност и стабилност, што го прави идеален за примена во различни сценарија.

Следно, го разгледавме процесот на замена на лица, кој вклучува детекција на значајни точки на лицата, користење на Делаунаова триангулација за дефинирање на обликот на лицето и примена на Афински трансформации за создавање на природен изглед. Примената на овие техники овозможува создавање на убедливи и визуелно добри резултати, со значителни примени во области како што се анимација, видеоигри и виртуелна реалност.

Имплементацијата на овие алгоритми во Python ја покажа моќта и флексибилноста на современите алатки за обработка на слики. Примената на библиотеките OpenCV и dlib значително го олеснува процесот на работа со слики и овозможува креирање на иновативни решенија за визуелни предизвици.

Заклучно, оваа семинарска работа не само што ја истражува теоретската основа на алгоритмите за детекција и замена на лица, туку и демонстрира практична имплементација и примена на тие алгоритми.

7. Референци

[1] Face swapping – Opencv with Python
[2] Face swapping using face landmark detection