Дигитално процесирање на слика

Тема:

Сегментација на стоматолошки панорамски рентген снимки

Лука Милошев – 221116

Ментор: Ивица Димитровски

1. Вовед

Историјата на стоматологијата е речиси исто толку древна како и историјата на човештвото и цивилизацијата, а најраните докази датираат од 7000 п.н.е. до 5500 п.н.е. Се смета дека стоматологијата била првата специјализација во медицината која продолжила да развива сопствена акредитирана диплома со свои специјализации. Стоматологијата честопати се подразбира и дека ја опфаќа сега во голема мера непостоечката медицинска специјалност на стоматологијата (проучување на устата и нејзините нарушувања и болести) поради што двата термина се користат наизменично во одредени региони.

Стоматологијата е една од гранките на медицината која постојано се развива со воведување на нови технологии за подобрување на дијагностиката и третманите. Еден од најзначајните предизвици во оваа област е точната и брза анализа на рендгенските снимки, кои се користат за откривање на кариес, инфекции, фрактури и други стоматолошки проблеми. Со развојот на вештачката интелигенција (AI) и компјутерската визија, автоматската обработка на медицински слики станува сè поефикасна и попрецизна. Сегментацијата на медицински слики е клучен процес во медицинската дијагностика бидејќи овозможува издвојување на релевантни анатомски структури, како што се забите на рендгенските снимки. Традиционалните методи за анализа на слики честопати се зависни од рачна обработка, што е време-јадно и подложно на грешки. Затоа, автоматизираните методи базирани на длабоко учење, како што е U-Net моделот, значително го подобруваат овој процес. Во оваа семинарска работа, ќе имплементираме U-Net модел за сегментација на стоматолошки рендгенски снимки, користејќи TensorFlow и Keras. Овој модел е специјално дизајниран за задачи со сегментација и се покажа како ефикасен во различни медицински апликации.

1.1. Важноста на рендгенската дијагностика во стоматологијата

Рендгенските снимки се суштински дел од современата стоматолошка пракса. Тие овозможуваат детално согледување на внатрешната структура на забите, корените, коскените ткива и можните патолошки состојби. Со точна анализа на овие снимки, стоматолозите можат да детектираат:

- Кариес кој не е видлив со голо око
- Инфекции и воспаленија на корените
- Позиција на импланти и ортодонтски апарати
- Фрактури и оштетувања на забите

Традиционалната анализа на рендгенските снимки се базира на визуелна инспекција од страна на стоматолозите, што може да доведе до субјективни проценки и пропуштени

дијагнози. Ова е причината зошто автоматизирани методи базирани на вештачка интелигенција стануваат сè попопуларни.

1.2. Улогата на вештачката интелигенција во медицината

Вештачката интелигенција (AI) значително го трансформира медицинскиот сектор, особено во анализата на медицински слики. Со користење на **длабоки невронски мрежи (Deep Learning)**, AI моделите можат да:

- Идентификуваат абнормалности на медицинските слики со висока точност
- Автоматизираат сложени дијагностички процеси
- Намалат човечки грешки и го забрзаат донесувањето на одлуки

Еден од најуспешните AI модели за анализа на медицински слики е **U-Net**, кој е специјално дизајниран за задачи со сегментација. Овој модел овозможува автоматско издвојување на анатомски структури, како што се забите на рендгенските снимки, што значително го олеснува процесот на дијагностика.

1.3. Важноста на информатиката во стоматологијата

Современата стоматологија не може да се замисли без употреба на информатички технологии. Од дигитални рендгенски системи до 3D моделирање и анализа на податоци, информатиката игра клучна улога во подобрување на стоматолошките услуги. Некои од главните придобивки вклучуваат:

- Подобрување на прецизноста на дијагнозите преку компјутерска анализа
- Автоматско водење на пациентски досиеја и дигитални записи
- 3D печатење на протези, импланти и ортодонтски апарати
- Телемедицина и далечинска консултација со стоматолози

2. Користење на библиотеки и нивна функционалност

- Matplotlib - е библиотека за програмскиот јазик Python и неговата нумеричка математичка екстензија NumPy. Обезбедува објектно-ориентирано API за вградување на графики во апликации со користење на комплети алатки за GUI за општа намена како Tkinter, wxPython, Qt или GTK. Исто така, постои процедурален интерфејс „pylab“ заснован на машина за состојба (како OpenGL).
- Cv2 - е име за внес на модул за opencv-python. „import cv2“ ја дава библиотеката OpenCV во скриптата на Python, овозможувајќи пристап до неговите функции за компјутерска визија и обработка на слики.
- Numpy - е библиотека за програмскиот јазик Python, додавајќи поддршка за големи, повеќедимензионални низи и матрици, заедно со голема колекција на математички функции на високо ниво за работа на овие низи. Numpy е математичка и научна компјутерска библиотека со отворен код за програмски задачи на Python. Името NumPy е кратенка за Numerical Python. Библиотеката NumPy нуди колекција на математички функции на високо ниво, вклучувајќи поддршка за повеќедимензионални низи, маскирани низи и матрици.
- Tensorflow - е софтверска библиотека за машинско учење и вештачка интелигенција. Може да се користи за низа задачи, но се користи главно за обука и заклучување на невронски мрежи. Таа е една од најпопуларните рамки за длабоко учење, заедно со други како PyTorch и PaddlePaddle. Тоа е бесплатен софтвер со отворен код објавен под лиценцата Apache 2.0. TensorFlow може да се користи во широк спектар на програмски јазици, вклучувајќи ги Python, JavaScript, C++ и Java, олеснувајќи ја неговата употреба во голем број апликации во многу сектори.
- Keras - е библиотека со отворен код која обезбедува интерфејс на Python. Keras прво бил независен софтвер, потоа интегриран во библиотеката TensorFlow. Keras нуди разбирлив интерфејс за дефинирање и тренирање на модели со неколку линии код, идеален за длабоко учење. Keras исто така нуди и: Поддршка за различни типови на слоеви, лесно вчитување и обработка на податоци, автоматско компајлирање и оптимизирање. Keras е една од најпопуларните библиотеки за длабоко учење поради својата едноставност, флексибилност и брзина
- Os - е вграден модул за оперативен систем со методи за интеракција со оперативниот систем, како создавање датотеки и директориуми, управување со датотеки и директориуми, влез, излез, променливи на животната средина, управување со процеси.

3. Модел

Овој проект има за цел да развие модел за сегментација на стоматолошки панорамски рентген снимки. Користен е U-Net модел за сегментација, кој врши детекција и издвојување на одредени региони на сликите, базирајќи се на дадени примероци со нивните соодветни маски.

Моделот е трениран на датасет кој содржи слики од DentalPanoramicXrays/Images, а соодветните ground truth маски се сместени во Orig_Masks. Откако моделот ќе биде трениран, истиот може да биде тестиран врз нови слики за да се добие предвидената сегментациска маска.

3.1. U-Net

U-Net е архитектура на длабока свиткувачка невронска мрежа, специјално дизајнирана за задачи на сегментација на слики, особено во биомедицинската област. Првично претставена во 2015 година во трудот "U-Net: Convolutional Networks for Biomedical Image Segmentation".

Клучни карактеристики на U-Net:

- **U-образна архитектура:** Мрежата се состои од два главни дела: контрактилен пат (енкодер) и експанзивен пат (декодер), кои заедно формираат U-форма. Енкодерот ја намалува димензионалноста на сликата и извлекува значајни карактеристики, додека декодерот ја обновува оригиналната резолуција и создава сегментациска маска.
- **Симетрична структура:** Енкодерот и декодерот имаат симетрична структура, што овозможува прецизно комбинирање на просторните и контекстуалните информации.
- **Конкатенација на карактеристики:** Во текот на декодирањето, U-Net користи конкатенација за да ги комбинира карактеристиките од соодветните нивоа на енкодерот и декодерот, што овозможува подобра реконструкција на деталите во сликата.

Примена на U-Net:

Иако првично беше развиена за биомедицинска сегментација, U-Net нашла примена во различни области, вклучувајќи:

- **Сегментација на медицински слики:** Како што се мозочни МРИ, КТ снимки и хистопатолошки слики.

- **Сателитски снимки:** За класификација на земјиште и детекција на објекти.
- **Автономни возила:** За разбирање на околината и детекција на патишта и препреки.

Популарноста на U-Net се должи на нејзината способност да постигне висока прецизност дури и со ограничен број на означени податоци, што ја прави идеална за задачи каде што е тешко да се добијат големи количини на означени слики.

3.2. Архитектура на моделот

Моделот е базиран на U-Net архитектурата, која е специјално дизајнирана за задачи на сегментација на слики. Се состои од два дела:

1. **Енкодер (Encoder/Contracting Path)**
    - Се користат свиткувачки (convolutional) слоеви со ReLU активација и филтри со големина 3x3.
    - По секоја свиткувачка операција се применува MaxPooling за намалување на димензиите.
    - По секој MaxPooling слој, бројот на филтри се зголемува за да се извлечат покомплексни карактеристики.
2. **Декодер (Decoder/Expanding Path)**
    - Декодерот користи UpSampling2D за зголемување на резолуцијата на сликите.
    - По секој UpSampling, резултатите се спојуваат (concatenate) со соодветните feature maps од енкодерот.
    - Се применуваат нови Conv2D операции за да се добие финалната сегментација.
3. **Излезен слој**
    - Последниот Conv2D слој има еден филтер (1x1) со sigmoid активациона функција.
    - Ова резултира со бинарна маска, каде што секој пиксел има вредност 0 или 1.

3.3. Подготовка на податоци

Вчитување на податоци

Функцијата load_data() ги вчитува сликите и нивните маски, ги претвора во numpy низи и ги нормализира со делење на 255. Важно е да се осигура дека редоследот на сликите и маските е ист, бидејќи моделот учи од нивните соодветства.

Претходна обработка

- Сликите се претвораат во grayscale бидејќи не е потребна боја.
- Големината на сликите се менува на 256x256 пиксели за да се усогласи со архитектурата на моделот.
- Пикселите се нормализираат (/255.0) за подобра конвергенција на моделот.

3.4. Тренинг на моделот

Моделот се компајлира со:

- Оптимизатор: Adam со learning_rate=0.001 – автоматски ја прилагодува стапката на учење.
- Функција за загуба: binary_crossentropy – користена затоа што имаме бинарна сегментација.
- Метрика: accuracy – следи точноста на предвидените пиксели.

Процес на тренирање

Моделот се тренира со fit() методот, каде:

- Тренинг сетот (X_train, y_train) се користи за учење.
- Валидација (validation_split=0.2) од 20% од тренинг податоците.
- Batch size е поставен на 8, што значи дека осум слики се обработуваат во исто време.
- Број на епохи: 50, што значи дека моделот ќе ги помине податоците 50 пати.

На крајот од тренингот, моделот се зачувува во dental_segmentation_model.h5.

3.5. Тестирање на моделот

Вчитување на тренираниот модел

Моделот се вчитува со:

model = load_model("dental_segmentation_model.h5")

Откако е вчитан, може да се тестира на нови слики.

Предвидување

1. Сликата се вчитува и претходно обработува на ист начин како при тренинг.
2. Моделот ја предвидува маската со model.predict().
3. Излезот е бинаризиран (> 0.5), со што се добива конечната сегментациска маска.

3.6. Заклучок

- Моделот е трениран со U-Net архитектурата, погодна за медицинска сегментација.
- Преку бинарна маска, ги детектира саканите области на панорамските рентген снимки.
- Се тестира на нови слики, при што дава сегментациски резултати.
- Визуелизацијата на резултатите овозможува брза проверка на точноста на предвидените маски.

4. Детална анализа на кодот за сегментација на стоматолошки панорамски рентген снимки. Примена на моделот

Овој код е наменет за вчитување на претходно трениран модел за сегментација и негова примена врз нови стоматолошки панорамски рентген снимки. Главната цел е автоматско издвојување на специфични структури од рентген сликите, користејќи машинско учење. Кодот вклучува претходна обработка на сликите, вчитување на моделот, предвидување на сегментациски маски и визуелизација на резултатите.

4.1. Обработка на податоците

Вчитување на сликата

- Кодот користи load_img() од Keras за вчитување на сликата во grayscale (црно-бело).
- Големината на сликата се менува на 256x256 пиксели за усогласеност со влезниот слој на моделот.

image = load_img(image_path, color_mode="grayscale", target_size=(256, 256))

Конверзија во numpy низа

- Сликата се претвора во нумеричка форма со img_to_array() и се нормализира со делење на 255.0.
- Додавање на batch димензија (expand_dims) за да може да биде обработена од моделот.

image = img_to_array(image) / 255.0

image = np.expand_dims(image, axis=0)

4.2. Вчитување на тренираниот модел

- Моделот се вчитува со load_model() од TensorFlow/Keras.
- Овој модел е веќе трениран со U-Net архитектура.

model = load_model("dental_segmentation_model.h5")

4.3. Процес на предвидување

4.3.1. Користење на моделот за предвидување

- Моделот прима обработена слика и враќа предвидена маска.

predicted_mask = model.predict(image)

4.3.2. Бинаризација на резултатот

- Излезот од моделот содржи вредности помеѓу 0 и 1.
- За да добиеме бинарна маска, сите вредности поголеми од 0.5 се поставуваат на 1, а останатите на 0.

predicted_mask = (predicted_mask > 0.5).astype(np.uint8)

4.4. Тестирање и евалуација

Мерење на точноста

- IoU (Intersection over Union): Мери преклопување помеѓу предвидената и вистинската маска.
- Dice Coefficient: Подобар индикатор за медицинска сегментација.

```
def dice_coefficient(y_true, y_pred):
    intersection = np.sum(y_true * y_pred)
    return (2. * intersection) / (np.sum(y_true) + np.sum(y_pred))
```

4.5. Визуелизација на резултатите

- Оригиналната слика и предвидената маска се прикажуваат една до друга.
- imshow() од matplotlib се користи за приказ.

```
fig, axs = plt.subplots(1, 2, figsize=(10, 5))
axs[0].imshow(original_image, cmap='gray')
axs[0].set_title("Original Image")
axs[1].imshow(predicted_mask.squeeze(), cmap='gray')
axs[1].set_title("Predicted Mask")
plt.show()
```

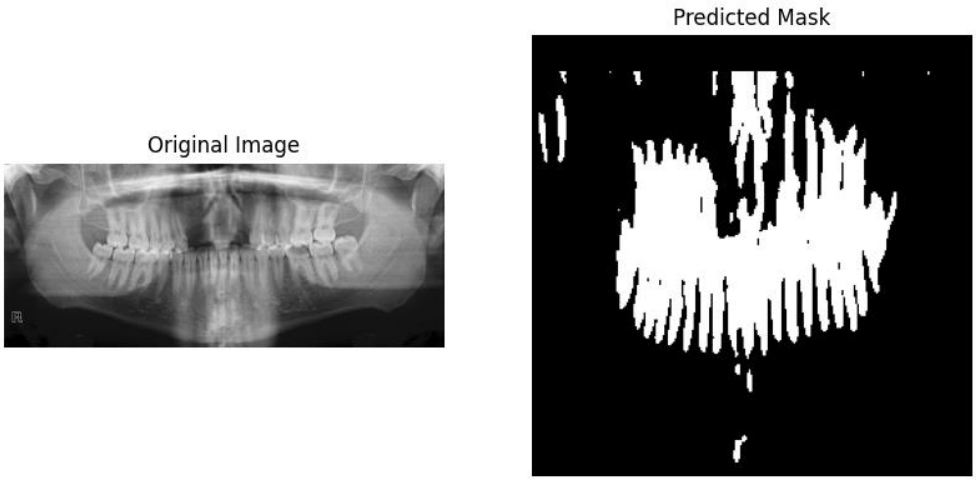
4.6. Заклучок

- Кодот успешно користи претходно трениран модел за сегментација на стоматолошки рентген снимки.
- Се спроведува претходна обработка, предвидување и визуелизација на резултатите.
- Мерењето на перформансите се врши преку IoU и Dice Coefficient.

Кодот нуди детално разбирање за функционирањето и неговата улога во автоматската анализа на стоматолошки рентген снимки.

5. Резултат од извршувањето на кодот

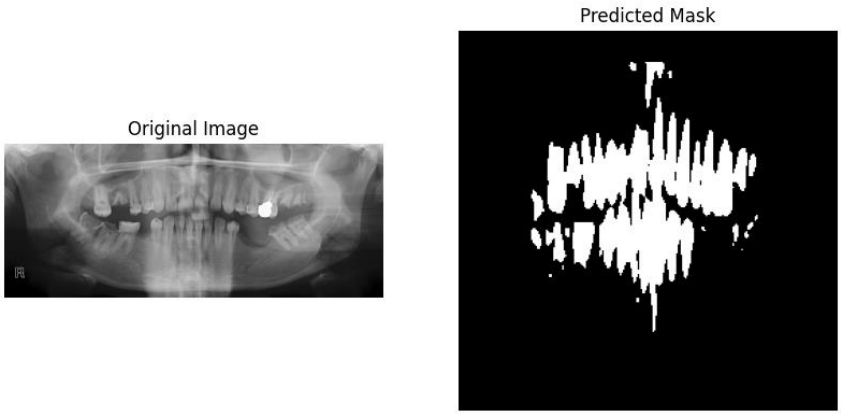
Моделот го тренираме во Keras, односно користиме функција која го прилагодува (тренинг) моделот според дадените влезни податоци и целни вредности.



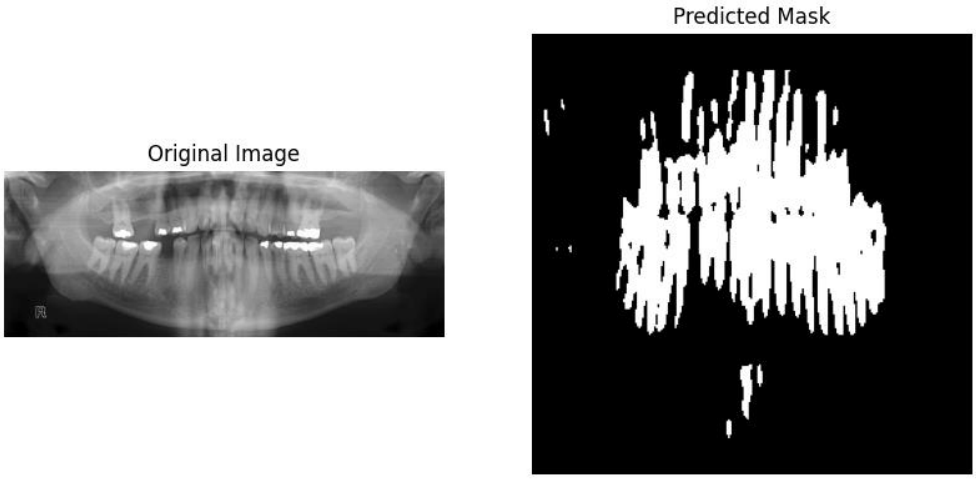
Слика бр.1, 50 епохи (повторување на тренингот)

Влезна слика(лево), од фолдерот кој што го имаме како низа од влезни податоци (рентгенски слики).

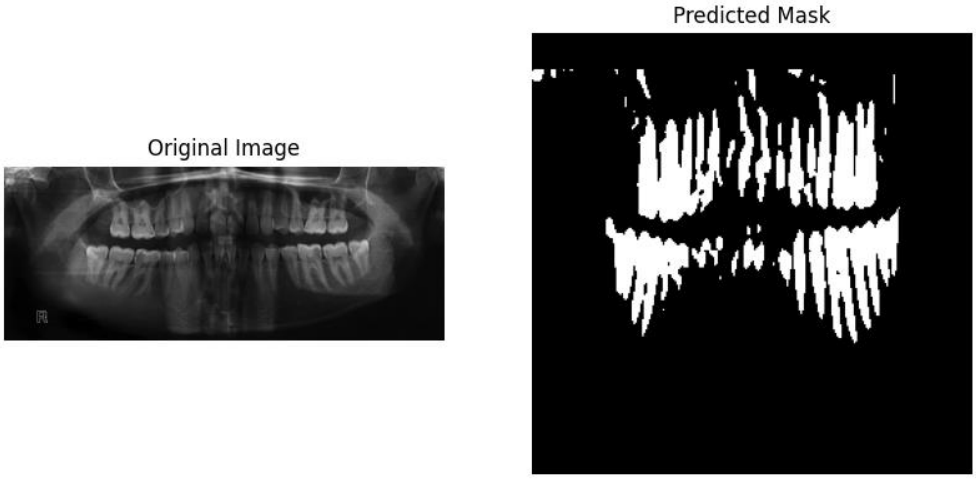
Излезна слика(десно), низа од очекувани резултати (сегментациски маски за секоја слика).



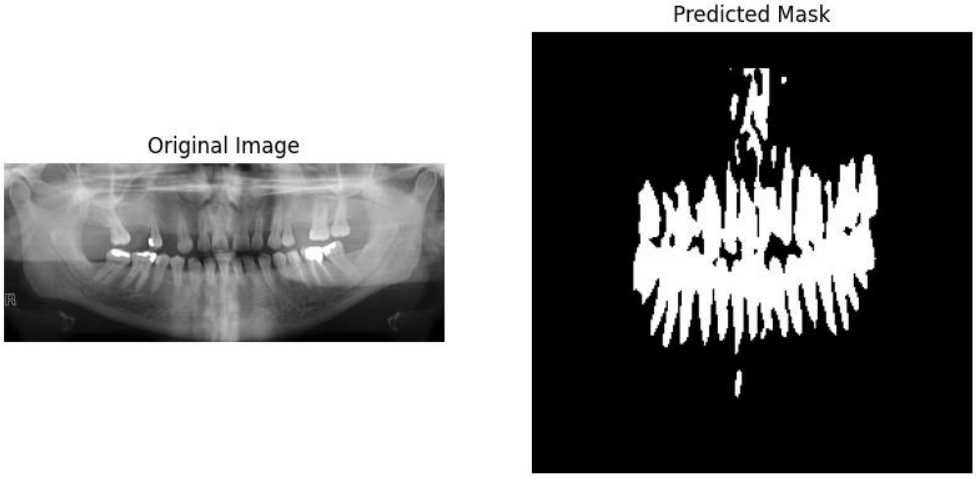
Слика бр.29, 50 епохи



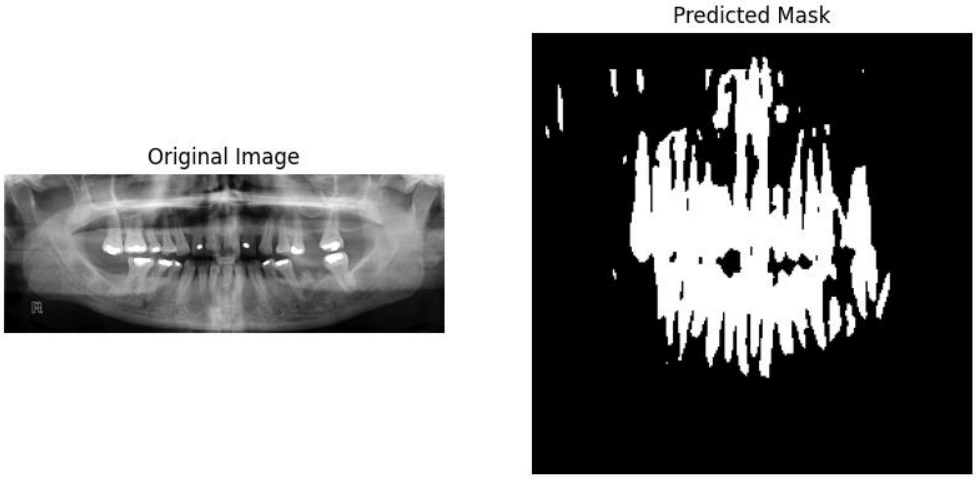
Слика бр.85, 50 епохи



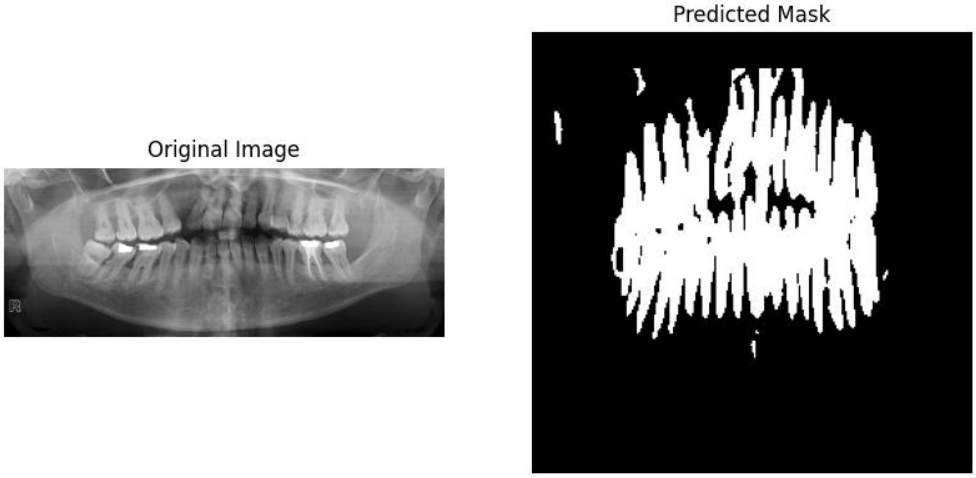
Слика бр.116, 50 епохи



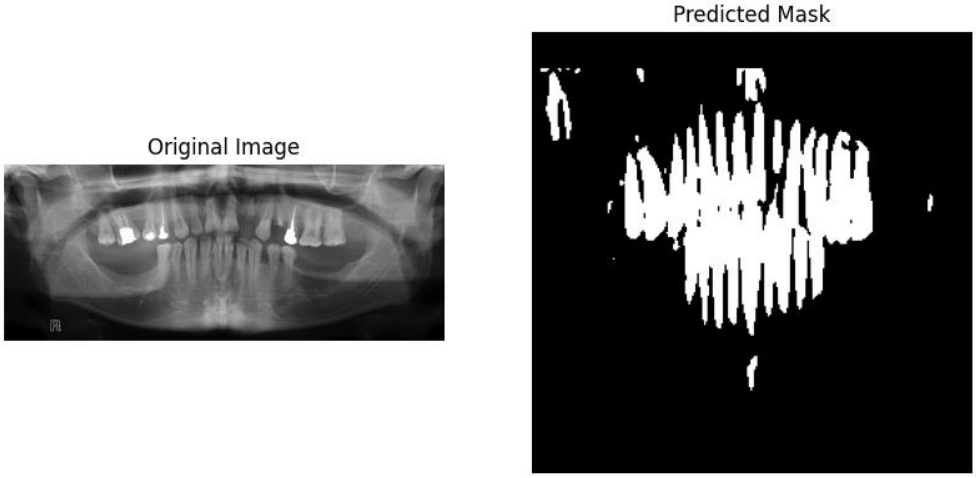
Слика бр.68, 50 епохи



Слика бр.71, 50 епохи



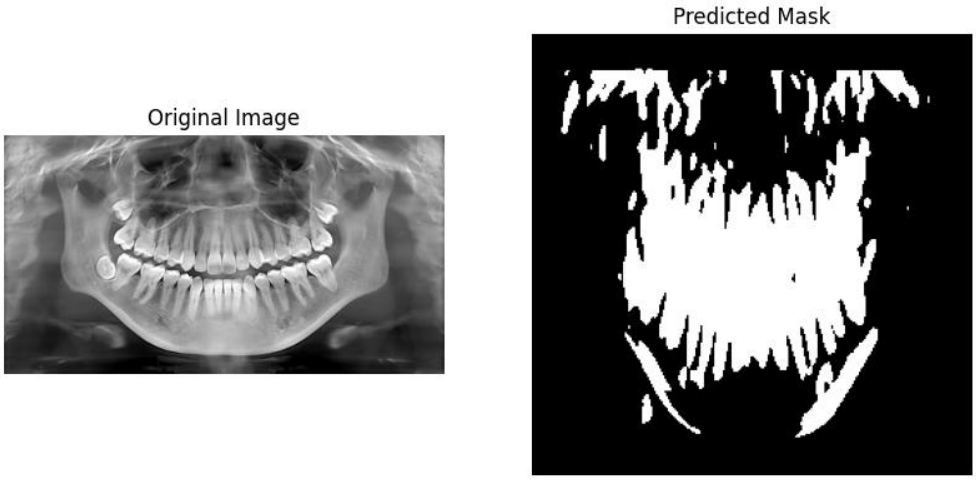
Слика бр.96, 50 епохи



Слика бр.103, 50 епохи

Покрај тестирање со слики на кои е направен моделот, може да се тестира на слики кои што се надвор од фолдерот кои што содржи слики за тестирање. Може да се внесе било која слика, но мора да се следат следните чекори: Да се внесе точната патека, да биде зачувана во правилен формат, да се скалира правилно и да се нормализира за да може моделот да ја разбере.

Еве пример на панорамска снимка која што не припаѓа на фолдерот со слики врз кој учеше и беше тестиран моделот.



Користена литература

https://pubmed.ncbi.nlm.nih.gov/32020135/

https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/

https://www.geeksforgeeks.org/u-net-architecture-explained/

https://en.wikipedia.org/wiki/Matplotlib

https://en.wikipedia.org/wiki/NumPy

https://en.wikipedia.org/wiki/TensorFlow

https://www.w3schools.com/python/module_os.asp#:~:text=Python%20has%20a%20built%2Din,set%20of%20methods%20and%20constants.

https://en.wikipedia.org/wiki/Keras