

Санкт-Петербургский Государственный Университет  
Факультет Прикладной Математики и Процессов Управления

**Отчет по практическому занятию №2**  
**“РЕАЛИЗАЦИЯ И ИССЛЕДОВАНИЕ ДЕТЕКТОРОВ ЛИЦ”**  
**в рамках курса “Прикладные задачи построения**  
**современных вычислительных систем”**

Выполнила студентка  
группы 17.Б12-пу  
Барташук Анастасия Евгеньевна

# Содержание

<b>Содержание</b>	<b>2</b>
<b>Цель работы</b>	<b>3</b>
<b>Template Matching</b>	<b>4</b>
<b>Fraunhofer</b>	<b>17</b>
<b>Viola-Jones</b>	<b>19</b>
<b>Линии симметрии лица</b>	<b>21</b>
<b>Тестирование на других лицах</b>	<b>24</b>
Template Matching	26
Viola-Jones	28
Линии симметрии лица	30
<b>Проверка работы Template Matching на шаблоне с другим лицом</b>	<b>32</b>
<b>Выводы</b>	<b>35</b>
<b>Заключение</b>	<b>36</b>

## Цель работы

Реализовать детектор лиц на основе метода Template Matching. Рассмотреть варианты детектора с различными шаблонами (целое лицо или его фрагменты). Исследовать качество детектирования лиц в различных условиях (освещение, его неравномерность, повороты/наклон лица, сокрытие части лица, расстояние от камеры).

Протестировать работу демоверсии детектора лиц «Fraunhofer», проверив возможности детектирования лица в различных условиях освещения, поворотов и движения головы.

Провести те же исследования для детектора Виолы-Джонса.

Определить линии симметрии лица – центральную и две локальных.

Сделать выводы по проделанной работе.

# Template Matching

Template Matching - это метод поиска и определения местоположения изображения, максимально похожего на заданный шаблон, на большом изображении. В OpenCV есть функция `cv2.matchTemplate()` для этой цели. Она просто перемещает изображение шаблона по входному изображению (как при двумерной свертке) и сравнивает шаблон и фрагмент входного изображения под изображением шаблона. В OpenCV реализовано несколько методов сравнения: `cv2.TM_CCOEFF`, `cv2.TM_CCOEFF_NORMED`, `cv2.TM_CCORR`, `cv2.TM_CCORR_NORMED`, `cv2.TM_SQDIFF`, `cv2.TM_SQDIFF_NORMED`. Они возвращают изображение в градациях серого, где каждый пиксель обозначает, насколько соседство этого пикселя совпадает с шаблоном.

Для нашего исследования использовались следующие картинки : идеальная фотография (Рисунки 1), фотография с плохим освещением (Рисунок 2), фотография с человеком в отдалении от камеры (Рисунок 3), фотография с наклоном головы (Рисунок 4), фотография с неравномерным светом (Рисунок 5), фотография с поворотом головы (Рисунок 6), фотография со скрытой частью лица (Рисунок 7).



Рисунок 1. “Идеальная” фотография



Рисунок 2. Фотография с плохим освещением



Рисунок 3. Удаленность от камеры



Рисунок 4. Наклон головы



Рисунок 5. Неравномерное освещение



Рисунок 6. Поворот головы



Рисунок 7. Скрыта часть лица

Из идеальной фотографии с помощью инструмента “Обрезать” в Paint создаем шаблон (Рисунок 8). Загружаем его и изначальную фотографию в программу и получаем результат (Рисунки 9-14). Как мы можем наблюдать, почти все методы справились с задачей определения лица по шаблону за исключением: `cv2.TM_CCORR`.



Рисунок 8. Шаблон



Рисунок 9. Результат `cv2.TM_SQDIFF`

Рисунок 10. Результат `cv2.TM_SQDIFF_NORMED`

Detected Point



Рисунок 11. Результат cv2.TM\_CCORR

Detected Point



Рисунок 12. Результат cv2.TM\_CCORR\_NORMED

Detected Point



Рисунок 13. Результат cv2.TM\_CCOEFF

Detected Point



Рисунок 14. Результат cv2.TM\_CCOEFF\_NORMED

Теперь проверим качество определения лиц с использованием шаблонов, полученных путем фрагментации первого шаблона: правая половина лица (Рисунок 15), глаз (Рисунок 16), нижняя половина лица (Рисунок 17).

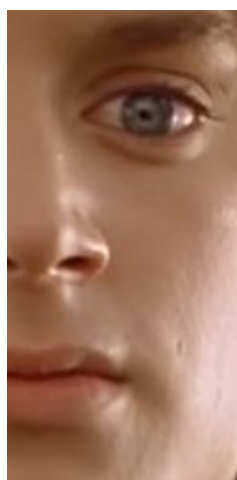


Рисунок 15.

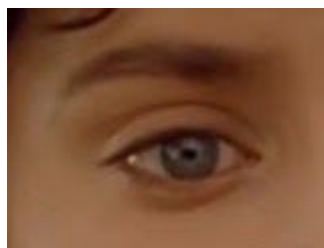


Рисунок 16.

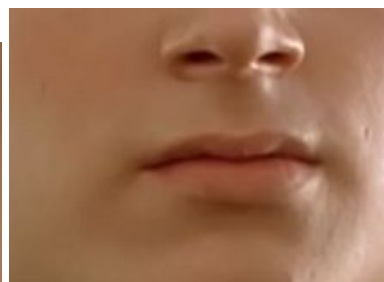


Рисунок 17.

Как мы видим по результатам работы нашего детектора, результаты работы с шаблоном “правая половина лица” не отличаются от результатов работы с изначальным шаблоном в методах cv2.TM\_CCORR, cv2.TM\_CCOEFF и cv2.TM\_CCOEFF\_NORMED, в

остальных - результат незначительно сдвинут вправо (Рисунки 18-20), результаты с шаблоном “глаз” менее точны и в основном сдвинуты влево (так как там и находится наш глаз) (Рисунки 21-23), результаты работы с шаблоном “нижняя половина лица” практически не отличаются от первых результатов (есть незначительный сдвиг вправо, но лицо все так же хорошо определяется) (Рисунки 24-26).



Рисунок 18. Шаблон “Правая половина лица”

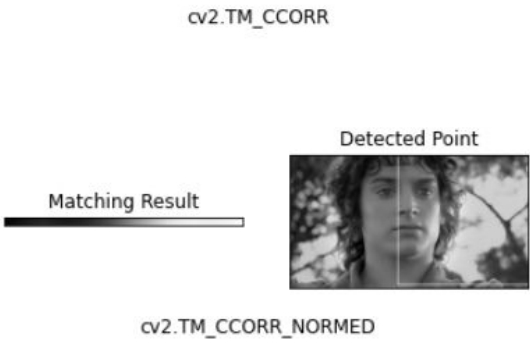


Рисунок 19. Шаблон “Правая половина лица”

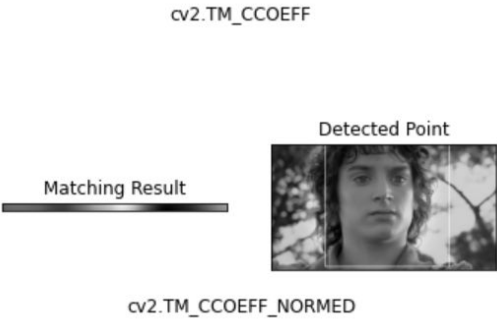


Рисунок 20. Шаблон “Правая половина лица”

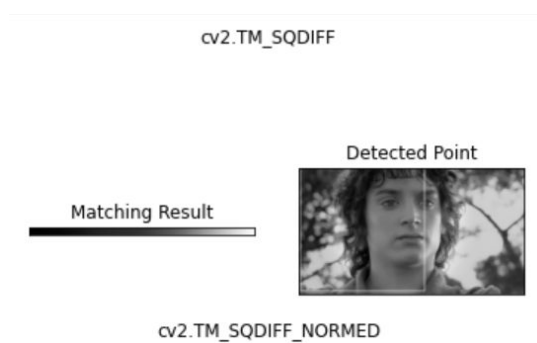


Рисунок 21. Шаблон “Глаз”

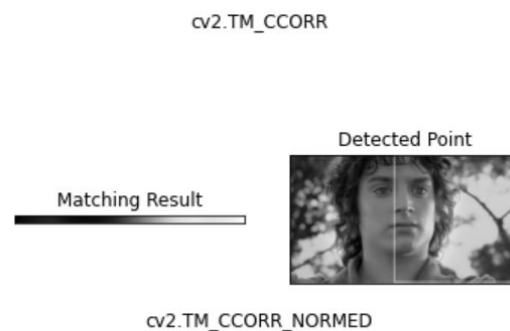


Рисунок 22. Шаблон “Глаз”



cv2.TM\_CCOEFF

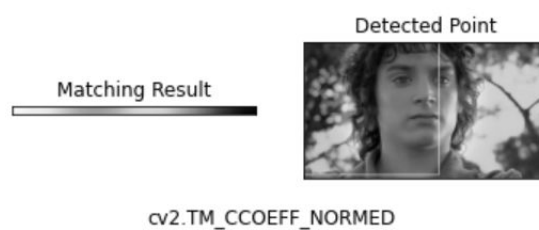


Рисунок 23. Шаблон “Глаз”



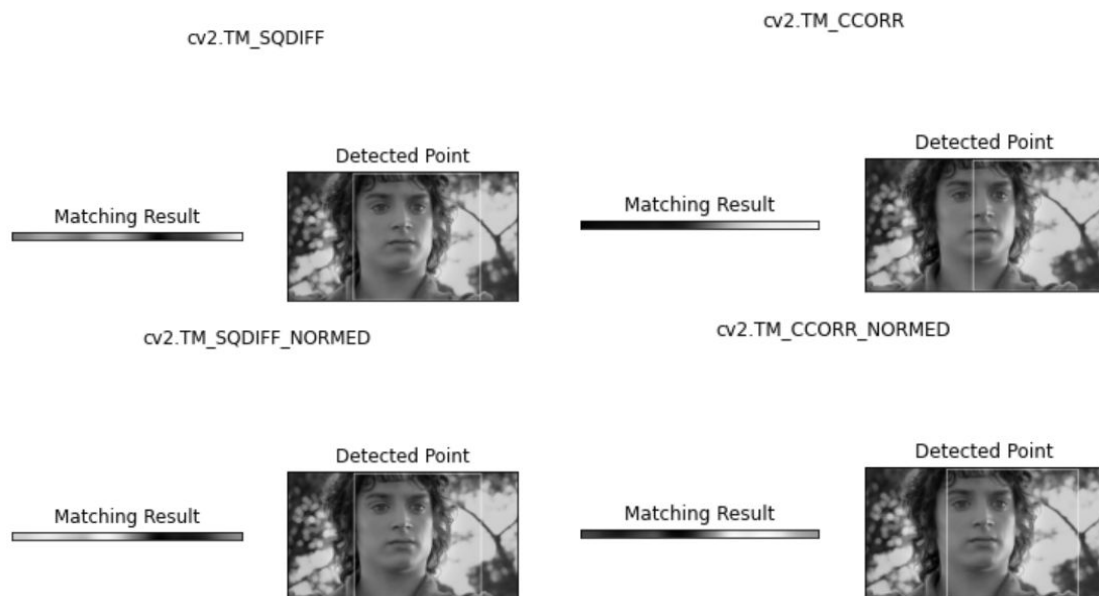


Рисунок 24. Шаблон “Нижняя половина лица” Рисунок 25. Шаблон “Нижняя половина лица”

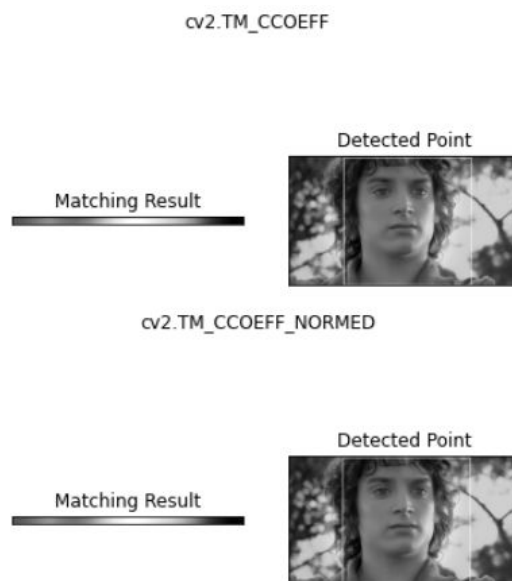


Рисунок 26. Шаблон “Нижняя половина лица”

Теперь проведем исследование качества детектирования лиц в различных условиях.

Как мы видим, при плохом освещении неплохо справились только первые три метода, остальные лица потеряли (Рисунки 27-29). При исследовании фотографии с неравномерным освещением лучше всего справился метод `cv2.TM_CCOEFF`, остальные либо вовсе потеряли лицо, либо указывали, что рука это лицо (Рисунки 30-32). При увеличении расстояния до камеры хорошо справился только один метод: `cv2.TM_CCORR_NORMED` (Рисунки 33-35). В случае, когда часть лица была скрыта,

справились три метода `cv2.TM_CCORR`, `cv2.TM_CCOEFF` и `cv2.TM_CCOEFF_NORMED` (Рисунки 36-38). При повороте головы хорошо себя показали методы `cv2.TM_CCORR`, `cv2.TM_CCOEFF` и `cv2.TM_CCOEFF_NORMED` (Рисунки 39-41). Когда же голова была наклонена, неплохо себя показал метод `cv2.TM_CCORR`, остальные - не справились с задачей (Рисунки 42-44).

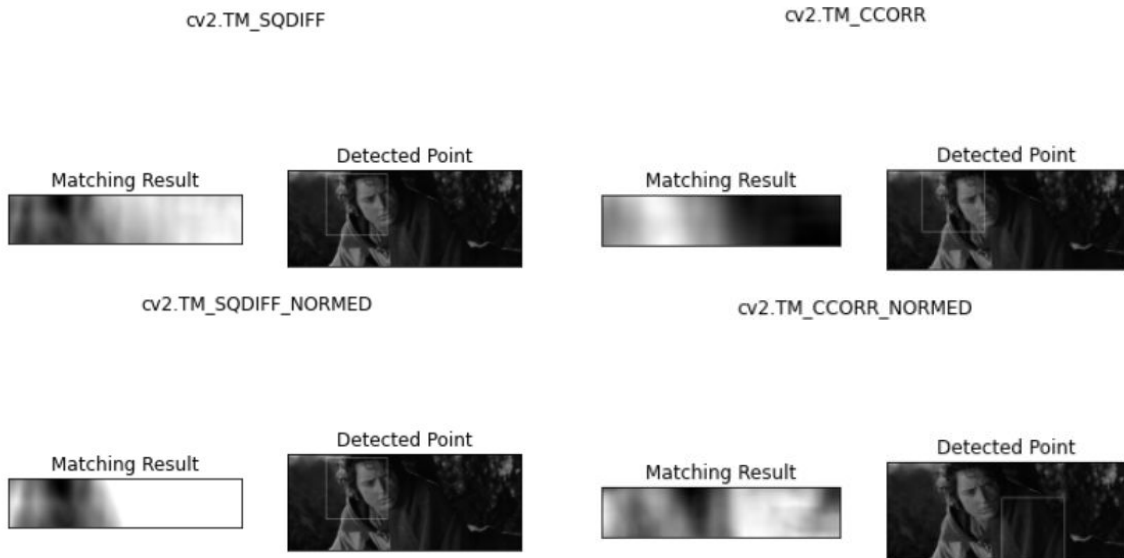


Рисунок 27. Методы при плохом освещении

Рисунок 28. Методы при плохом освещении

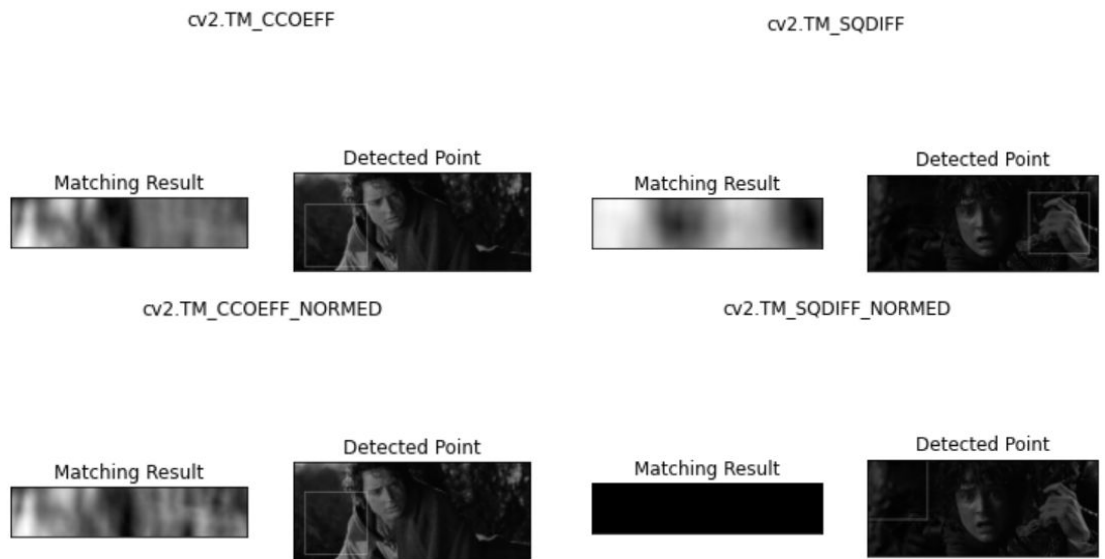


Рисунок 29. Методы при плохом освещении

Рисунок 30. Методы при неравномерном освещении

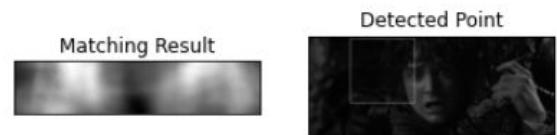
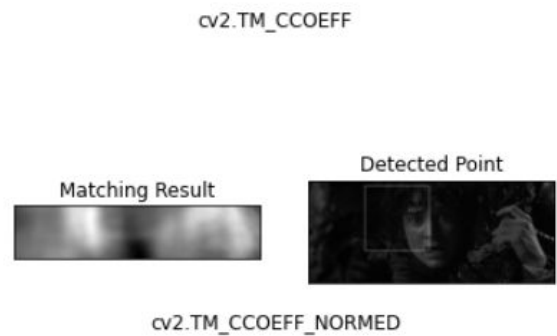
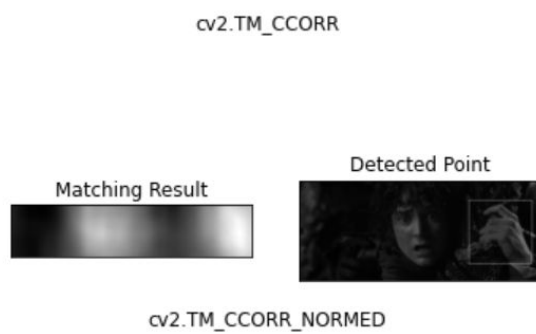


Рисунок 31. Методы при неравномерном освещении

Рисунок 32. Методы при неравномерном освещении

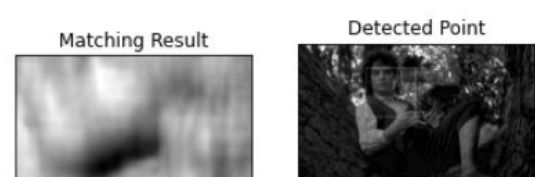
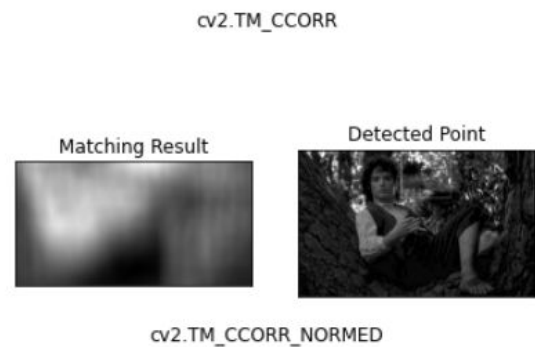
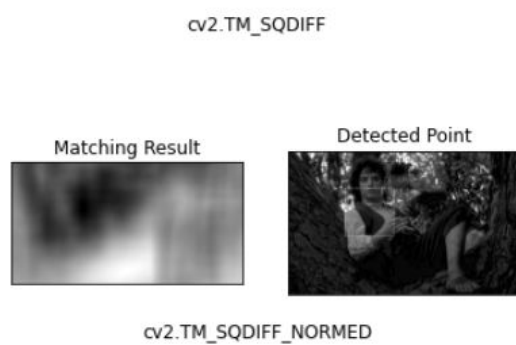
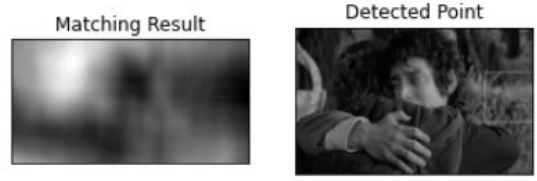
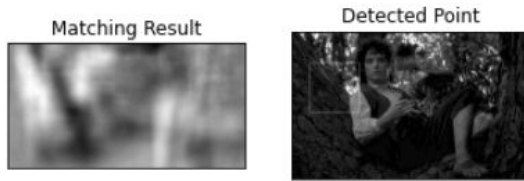


Рисунок 33. Методы при удалении от камеры

Рисунок 34. Методы при удалении от камеры

cv2.TM\_CCOEFF

cv2.TM\_SQDIFF



cv2.TM\_CCOEFF\_NORMED

cv2.TM\_SQDIFF\_NORMED

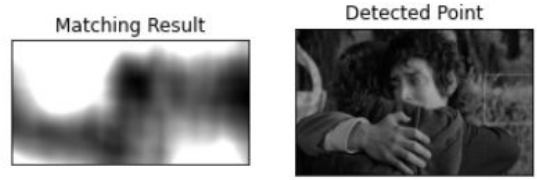
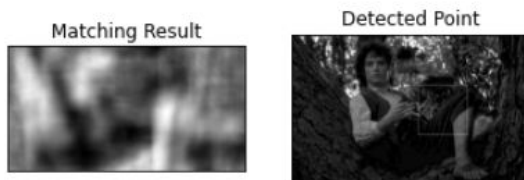


Рисунок 35. Методы при удалении от камеры

Рисунок 36. Методы при скрытой части лица

cv2.TM\_CCORR

cv2.TM\_CCOEFF



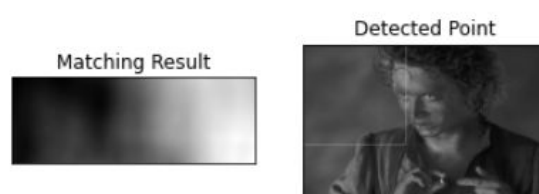
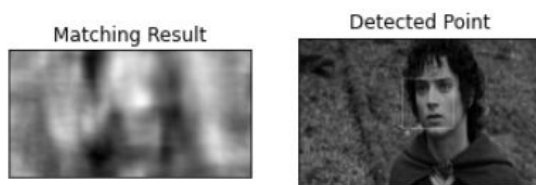
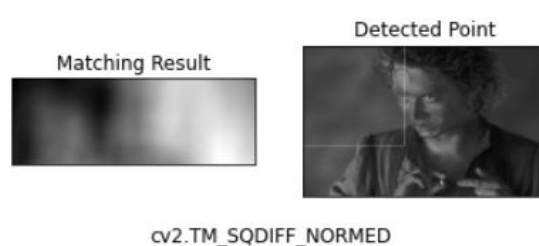
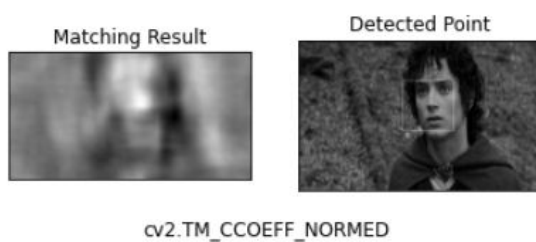
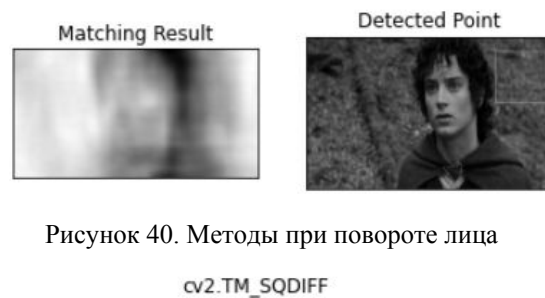
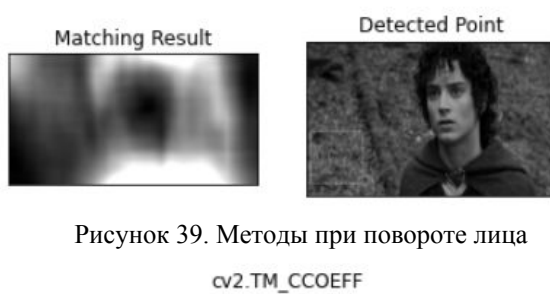
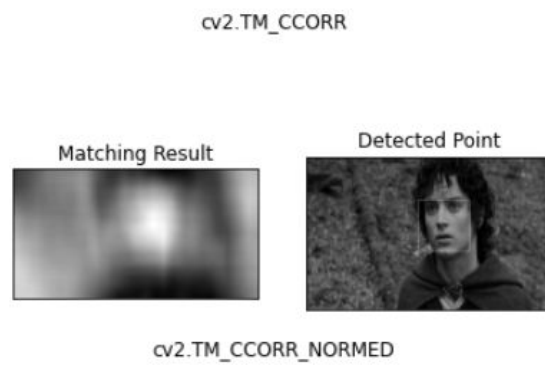
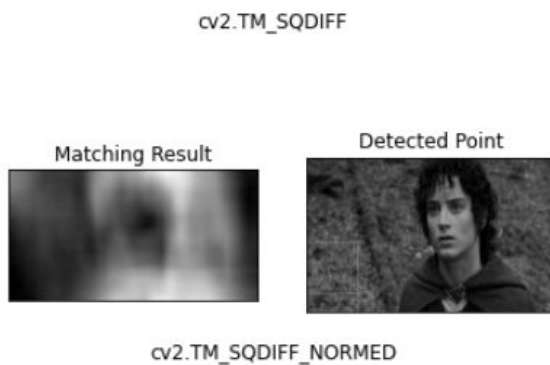
cv2.TM\_CCORR\_NORMED

cv2.TM\_CCOEFF\_NORMED



Рисунок 37. Методы при скрытой части лица

Рисунок 38. Методы при скрытой части лица



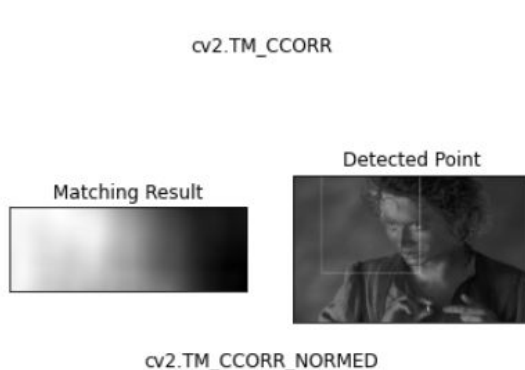


Рисунок 43. Методы при наклоне головы

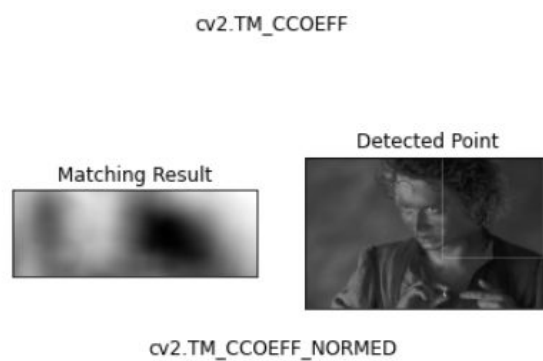


Рисунок 44. Методы при наклоне головы

Исходя из вышеперечисленных результатов, мы можем составить таблицу, оценивающую качество работы наших методов (Таблица 1).

	Ориг инал	Шабл он “прав ая полов ина лица”	Шабл он “глаз ”	Шабл он “ниж няя полов ина лица”	Плох ое осве щени е	Нера вном ерное осве щени е	Расст ояние до камер ы	Часть лица скрыт а	Пово рот голов ы	Накл он голов ы
cv2.T M_C COEF F	+	+	+/-	+	-	+/-	-	+/-	+	-
cv2.T M_C COEF	+	+	+/-	+	-	-	-	+	+	-

F_NO RME D										
cv2.T M_C COR R	-	-	+/-	-	+	-	-	+	+	+/-
cv2.T M_C COR R_N ORM ED	+	+	+	+	-	-	+	-	-	-
cv2.T M_S QDIF F	+	+	+/-	+	+	-	-	-	-	-
cv2.T M_S QDIF F_NO RME D	+	+	+/-	+	+	-	-	-	-	-

Таблица 1. Результаты работы методов Template Matching при различных условиях

Как мы видим, лучше всего справляются методы cv2.TM\_CCOEFF и cv2.TM\_CCOEFF\_NORMED, хуже всего - cv2.TM\_CCORR.

В большинстве случаев лица определяются верно, что говорит нам о достаточно качественной работе детектора. Однако при работе с фотографией с неравномерным освещением лучше всего применять метод `cv2.TM_CCOEFF`, при удаленности человека от камеры - `cv2.TM_CCORR_NORMED`, а при наклоне головы - `cv2.TM_CCORR`, также не стоит брать в качестве шаблона глаз или другой орган лица.



# Fraunhofer

Данный программный пакет представлен в нашей работе в качестве демо версии, но даже она отлично справляется с задачей.

При запуске программа автоматически подключается к веб-камере, однако есть возможность загрузки фотографий или видео и с компьютера.

Далее с помощью кнопок на верхней панели программы можно настраивать определение лица в целом, глаз, носа, губ, пола, возраста и эмоций.

Как мы видим, данное приложение справилось со своей задачей, правильно определив лица на всех фотографиях (Рисунки 45-51).

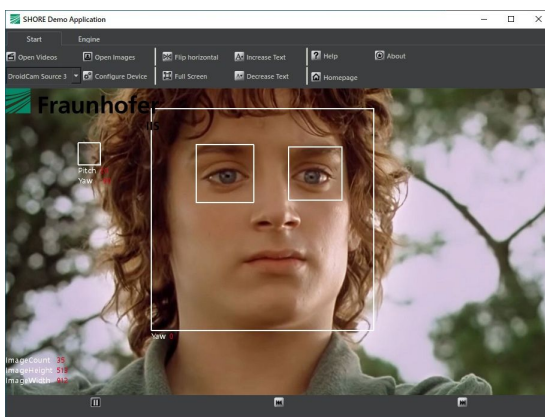


Рисунок 45. Результат работы Fraunhofer

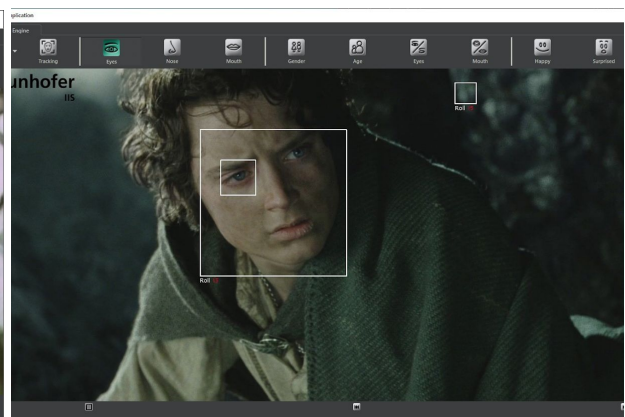


Рисунок 46. Результат работы Fraunhofer

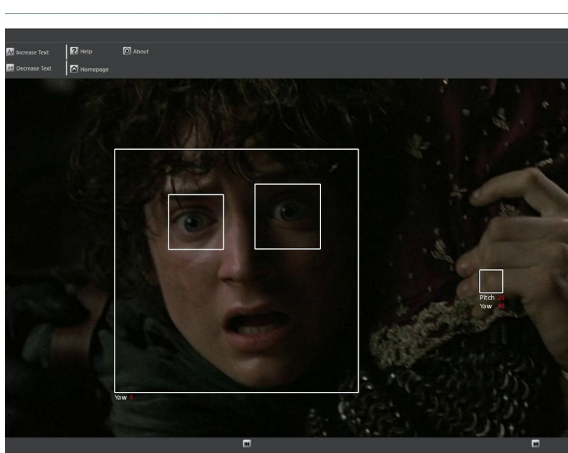


Рисунок 47. Результат работы Fraunhofer

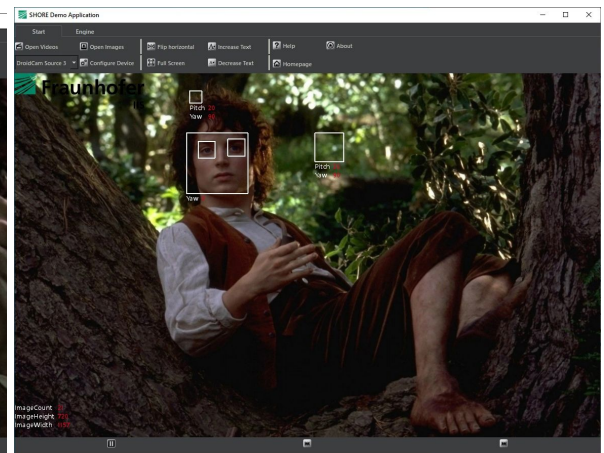


Рисунок 48. Результат работы Fraunhofer

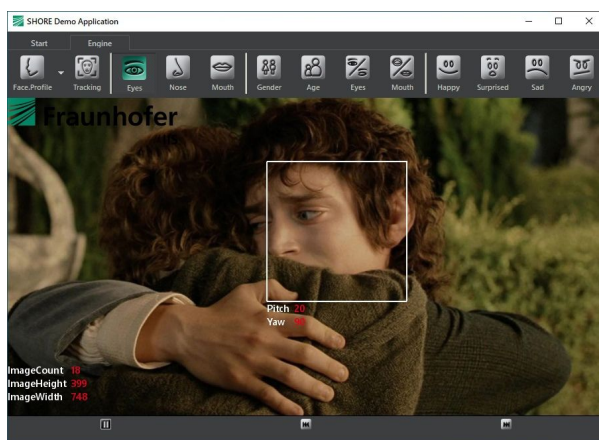


Рисунок 49. Результат работы Fraunhofer

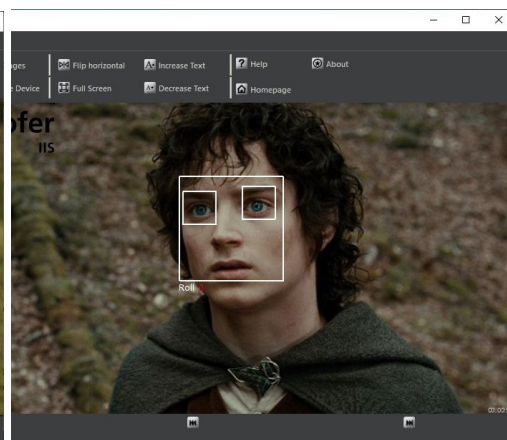


Рисунок 50. Результат работы Fraunhofer

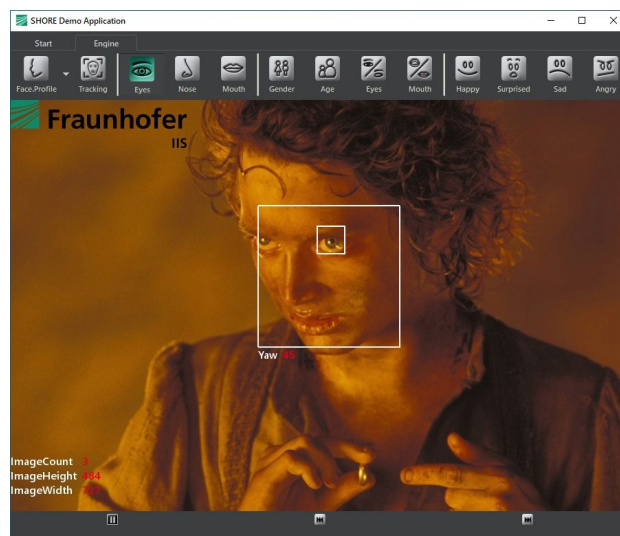


Рисунок 51. Результат работы Fraunhofer

# Viola-Jones

Теперь перейдем к рассмотрению детектора Виолы-Джонса.

Обнаружение объектов с использованием каскадных классификаторов Хаара - это эффективный метод обнаружения объектов, предложенный Полом Виолой и Майклом Джонсом в их статье «Быстрое обнаружение объектов с использованием усиленного каскада простых функций» в 2001 году. Это подход основан на машинном обучении. Каскадная функция обучается на большом количестве положительных и отрицательных изображений. Затем она используется для обнаружения объектов на других изображениях. Данная функция входит в пакет OpenCV и включает в себя определение не только самого лица, но и его объектов (глаз, рта и тд). Нам же для работы понадобится только файл `haarcascade_frontalface_default.xml`, он и поможет нам определить на картинке лицо. Результаты представлены на рисунках 52-54.

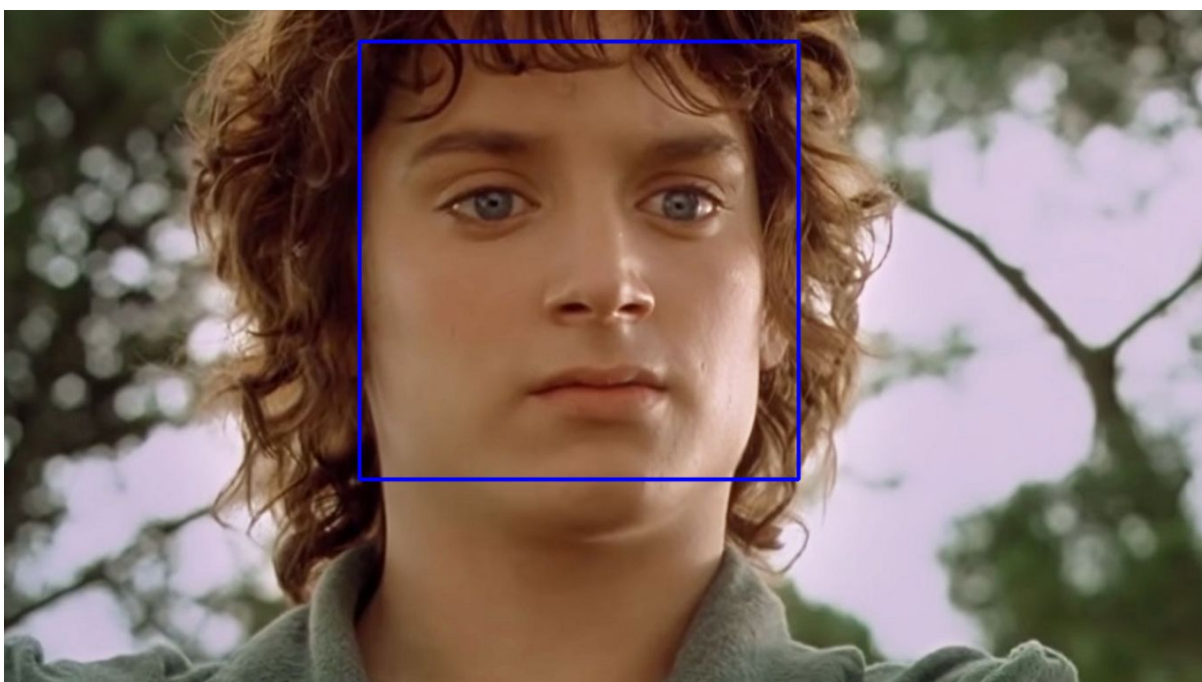


Рисунок 52. Результат работы алгоритма Viola-Jones для “идеальной” фотографии



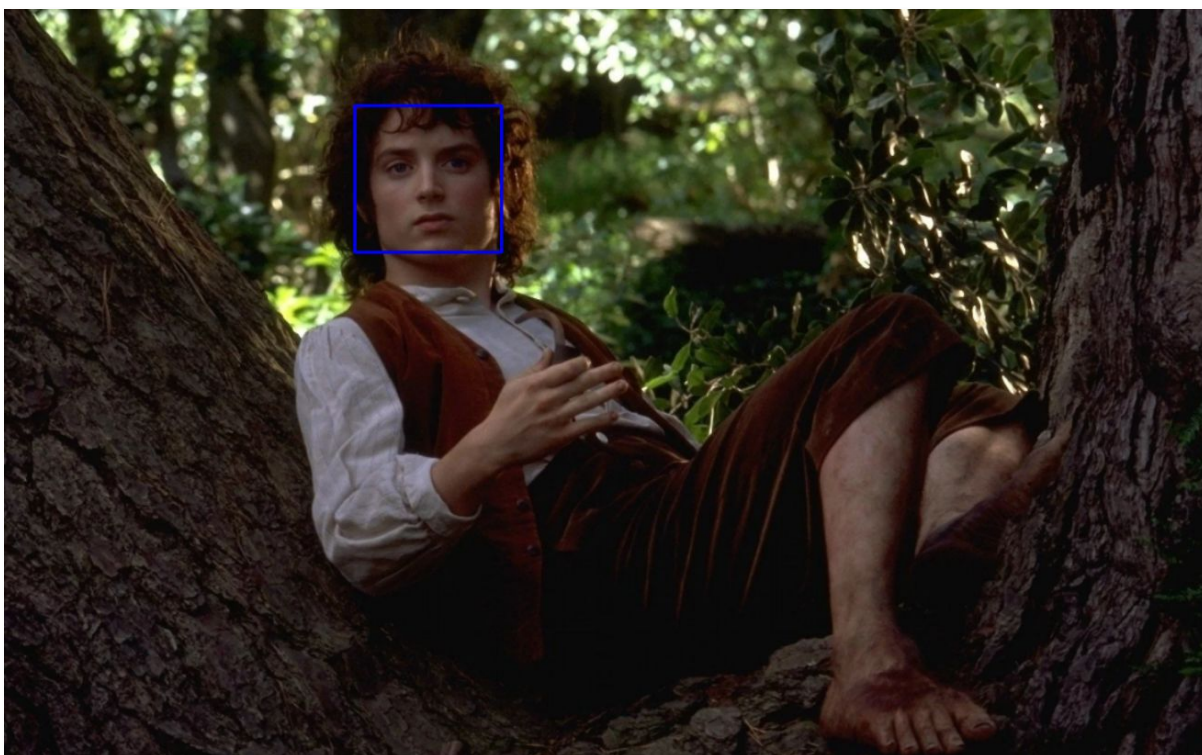


Рисунок 53. Результат работы алгоритма Viola-Jones при удаленности человека от камеры

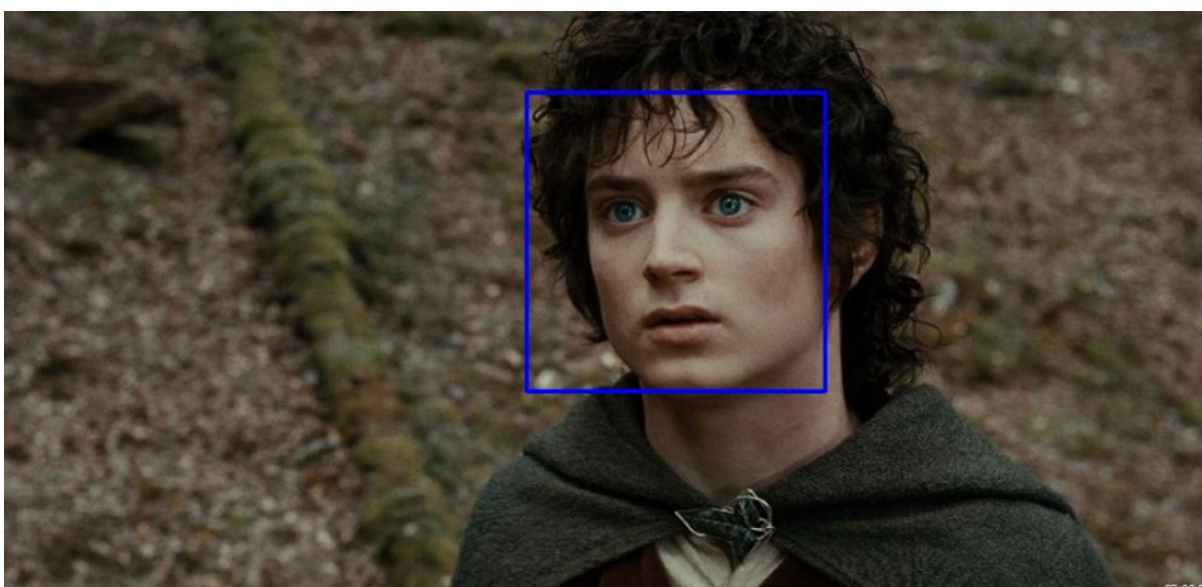


Рисунок 54. Результат работы алгоритма Viola-Jones при повороте лица

Как мы видим, данный детектор смог определить лица не на всех фотографиях. При наклоне головы, плохом и неравномерном освещении или сокрытии части лица программа не смогла справиться с задачей.

## Линии симметрии лица

Данный метод также основан на детекции лица. После определения лица расставляются опорные точки (всего их 68) с помощью пакета dlib на месте глаз, бровей, лица, носа и нижней челюсти. Далее уже на основе полученной разметки строятся линии симметрии лица: центральная - через точки 28 и 9 (переносица и середина подбородка), локальные - параллельно центральной на расстоянии до зрачков глаз.

Данный метод также не смог справиться идеально: лицо в наклоне и отчасти скрытое были не определены (Рисунки 55-59).

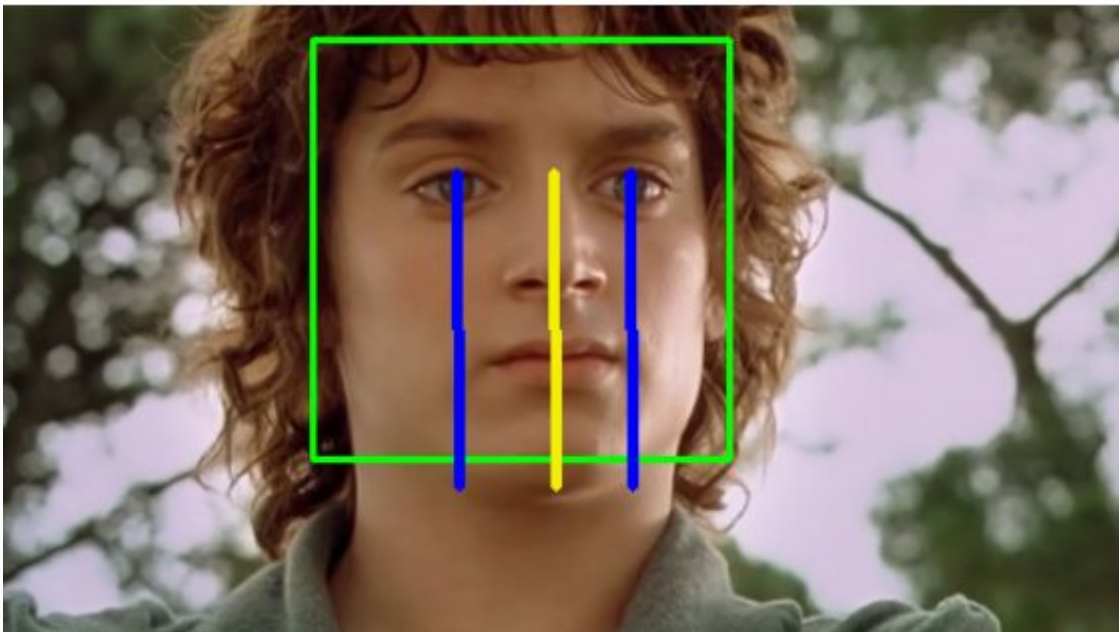


Рисунок 55. Линии симметрии на “идеальной” фотографии



Рисунок 56. Линии симметрии при плохом освещении



Рисунок 57. Линии симметрии при удалении от камеры



Рисунок 58. Линии симметрии при неравномерном освещении



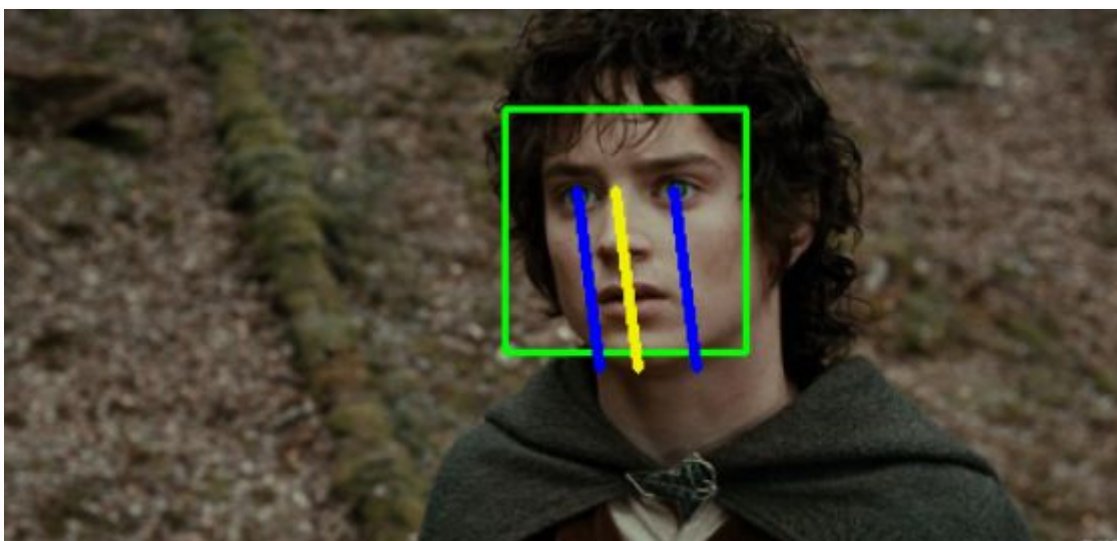


Рисунок 59. Линии симметрии при повороте головы

## Тестирование на других лицах

Теперь протестируем работу наших методов на других лицах, представленных на рисунках 60-66.



Рисунок 60. Эталонный



Рисунок 61. При плохом освещении

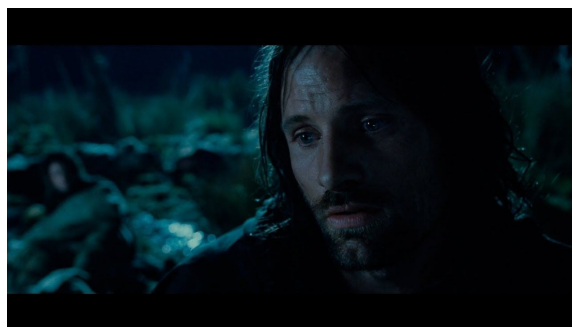


Рисунок 62. При неравномерном освещении



Рисунок 63. Удаление от камеры



Рисунок 64. Поворот головы



Рисунок 65. Наклон головы



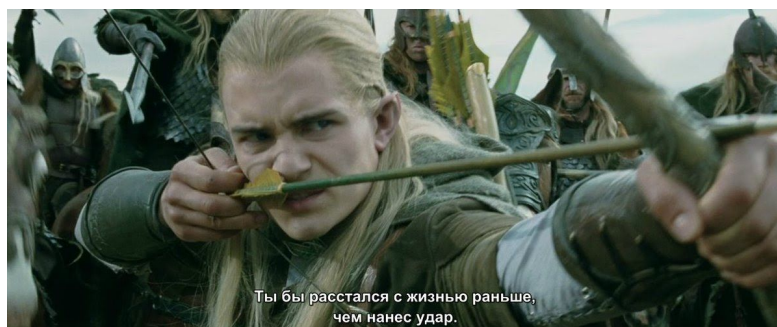


Рисунок 66. Частично скрытое лицо

## Template Matching

Данный метод справился неплохо, отчасти не осилив только плохое и неравномерное освещение и поворот головы. Результаты представлены на рисунках 67-73.

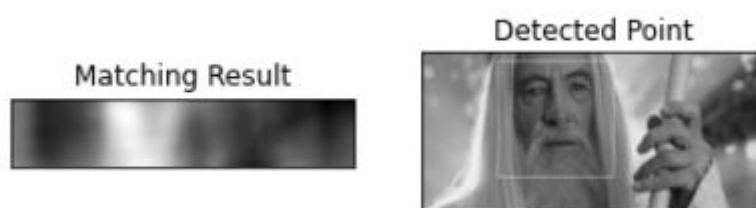


Рисунок 67. Результат работы для эталонной фотографии

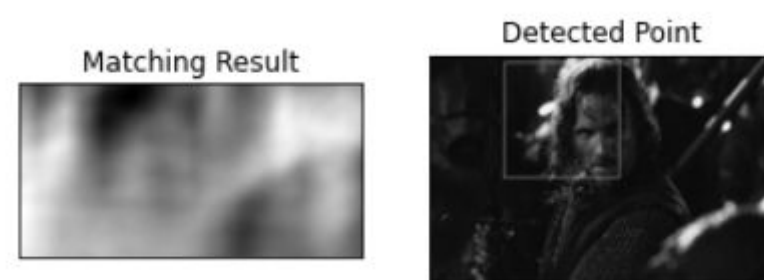


Рисунок 68. Результат работы при плохом освещении

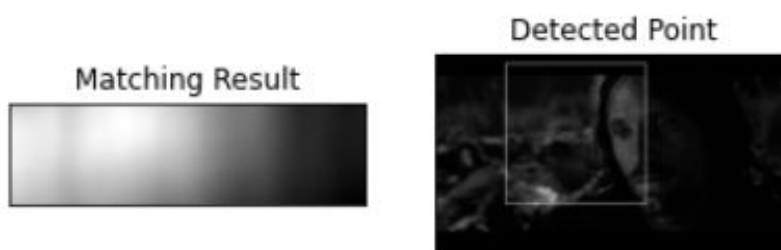


Рисунок 69. Результат работы при неравномерном освещении

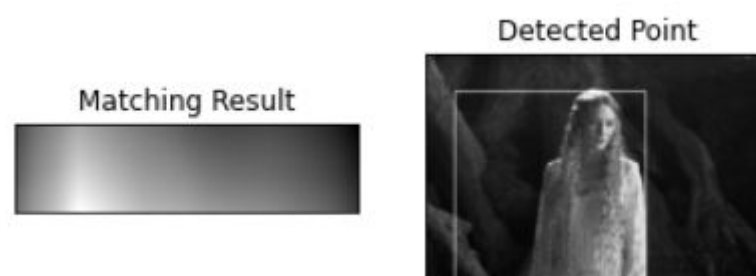


Рисунок 70. Результат работы при удалении от камеры



Рисунок 71. Результат работы при сокрытии части лица

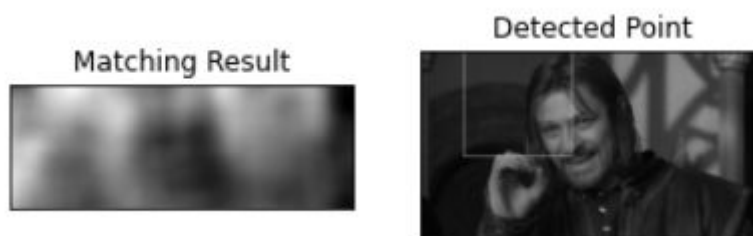


Рисунок 72. Результат работы при повороте лица

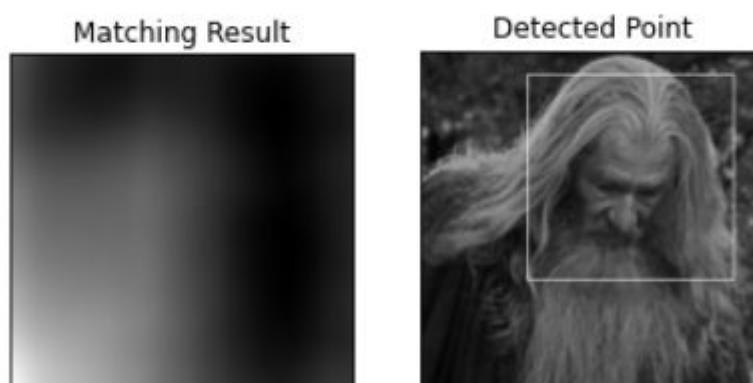


Рисунок 73. Результат работы при наклоне головы

## Viola-Jones

Метод обнаружение объектов с использованием каскадных классификаторов Хаара смог определить лица везде кроме случаев плохого освещения и наклона головы (Рисунки 74-77). Также в случае с анфасом при наличии бороды метод не обнаружил лицо, поэтому фотография была заменена на фотографию с рисунка 78 (рисунок уже с результатом).

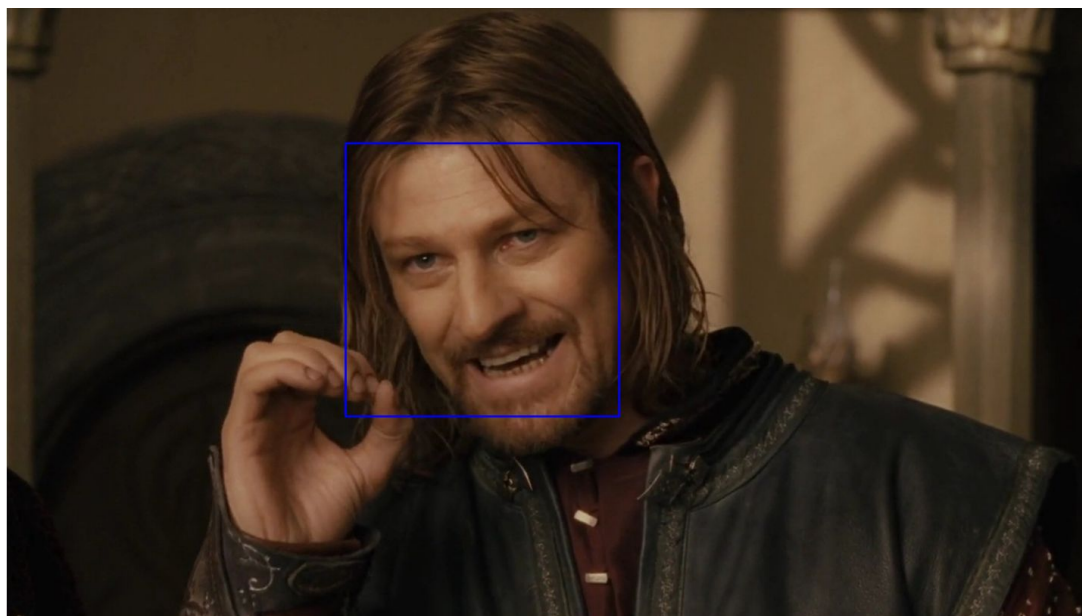


Рисунок 74. Результат при повороте головы

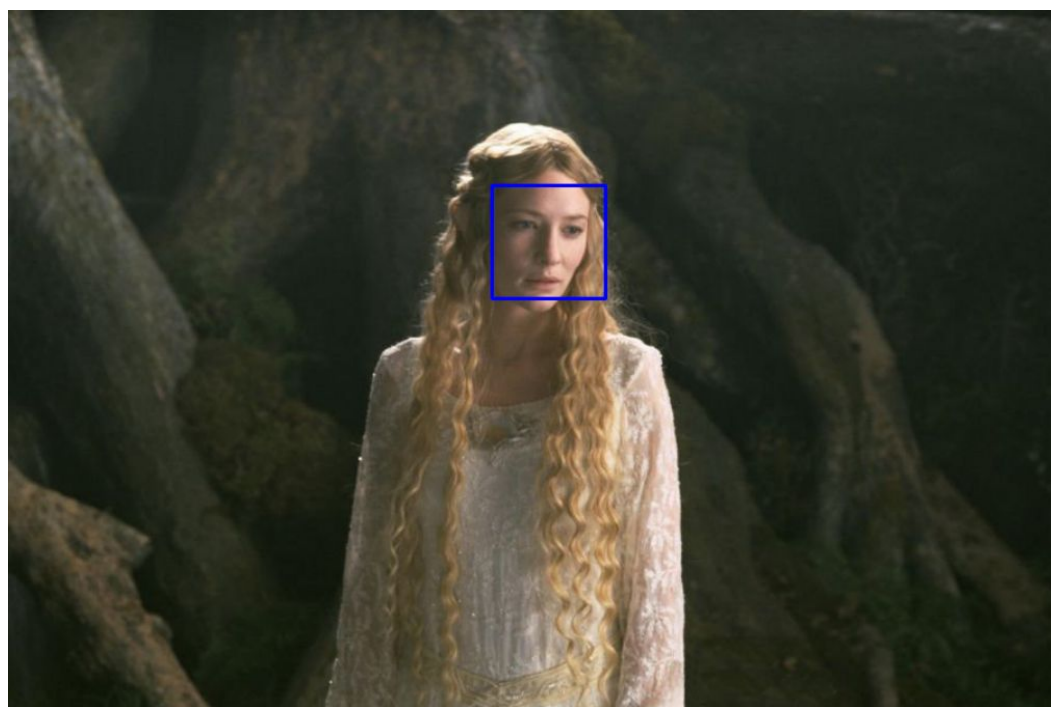


Рисунок 75. Результат при удалении от камеры

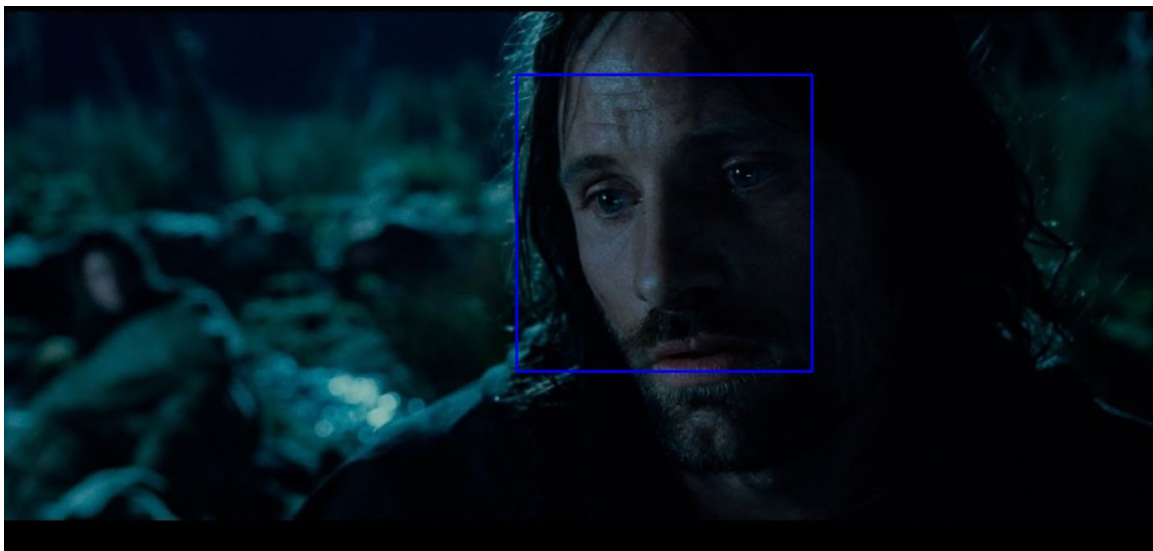


Рисунок 76. Результат при неравномерном освещении

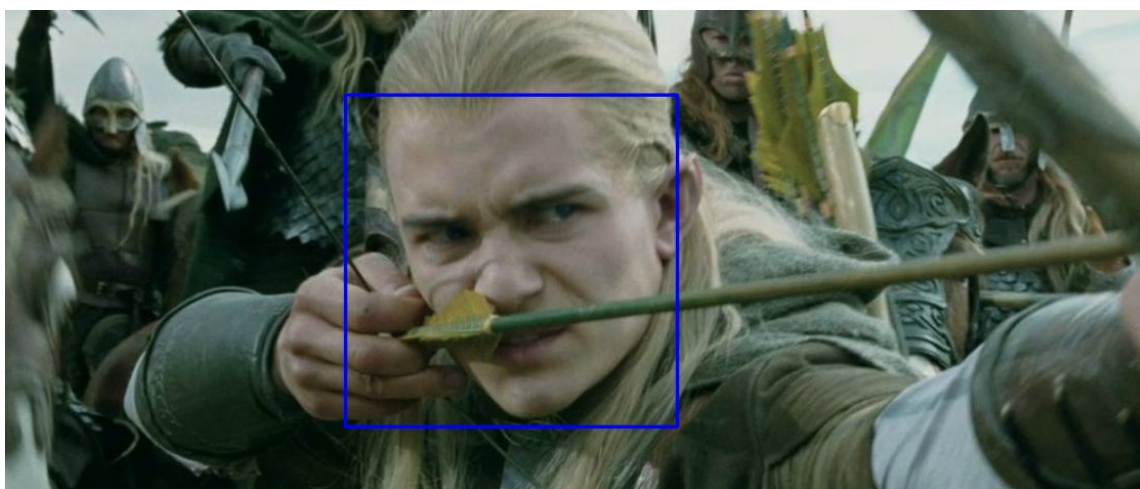


Рисунок 77. Результат при частично скрытом лице

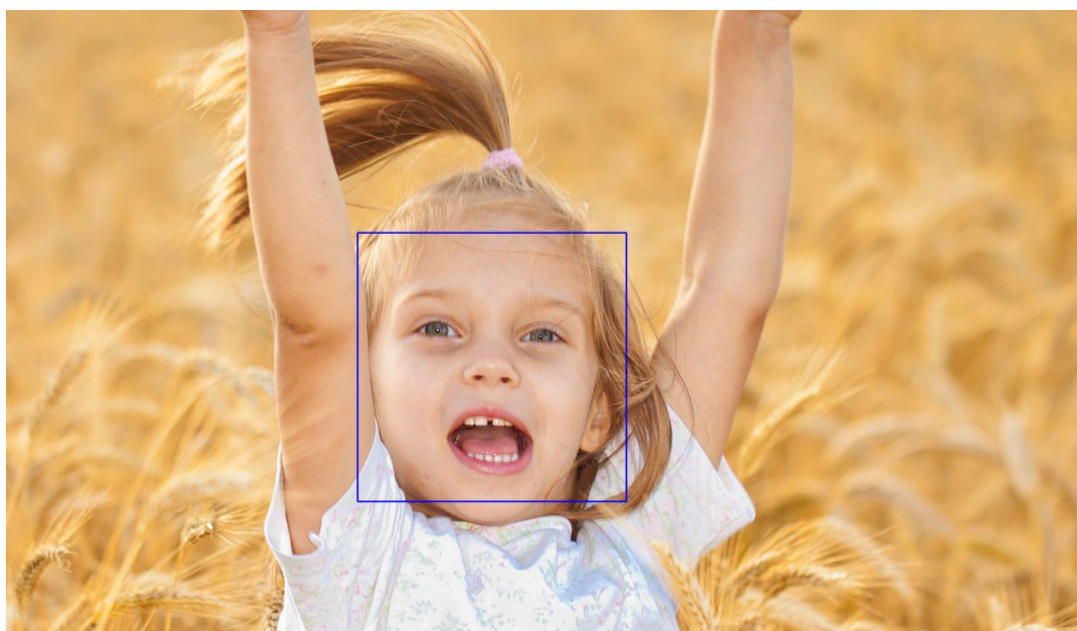


Рисунок 78. Результат для эталонной фотографии



## Линии симметрии лица

Детектор лица, работающий с помощью библиотеки dlib, также смог справиться далеко не со всем: ему не дали случаи плохого и неравномерного освещения, частичного сокрытия лица и наклона головы. В остальном метод справился со всеми остальными случаями (Рисунки 79-81).

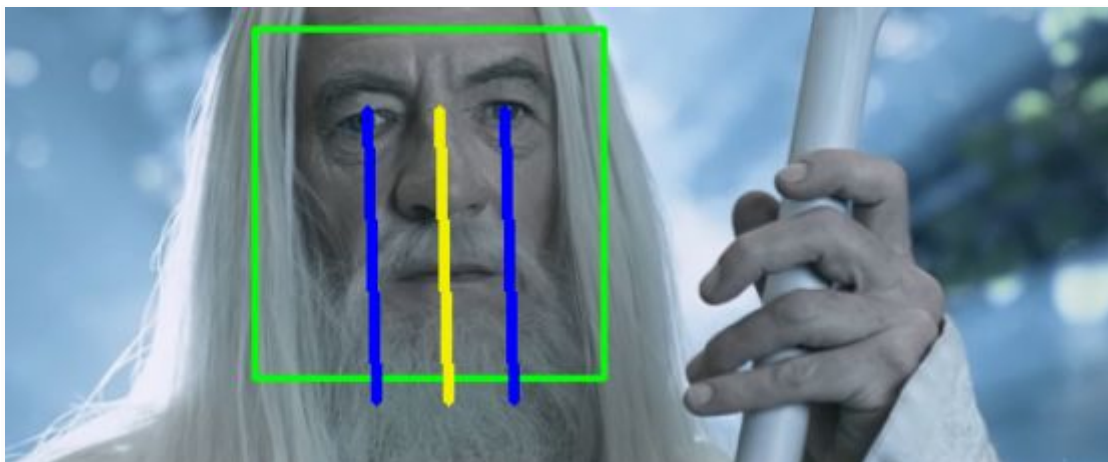


Рисунок 79. Линии симметрии для эталонной фотографии



Рисунок 80. Линии симметрии при удалении от камеры

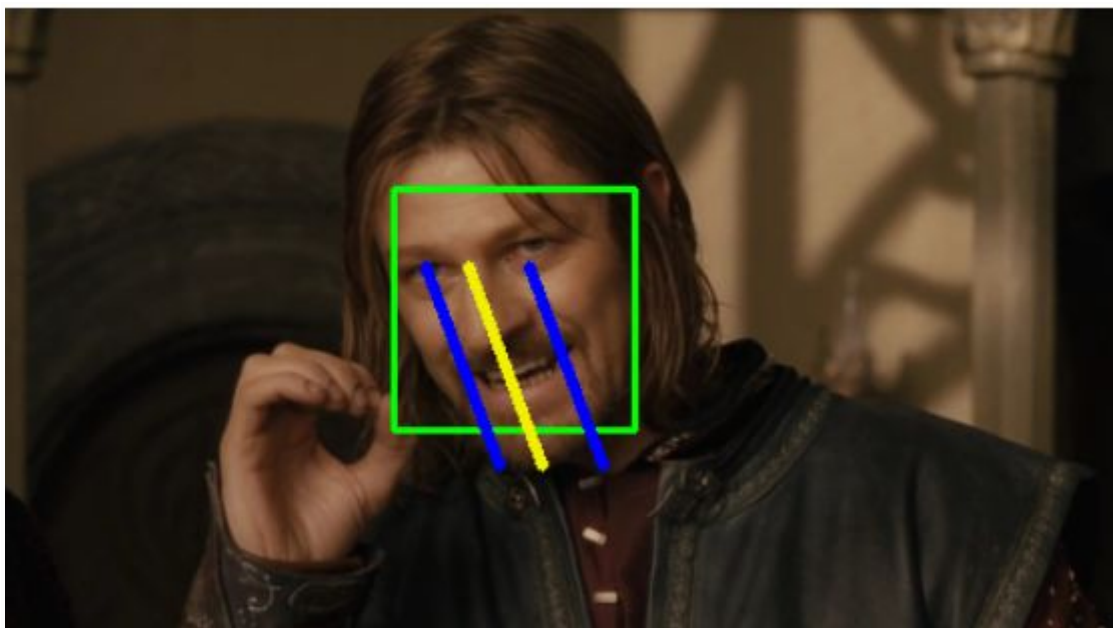


Рисунок 81. Линии симметрии при повороте лица.

## Проверка работы Template Matching на шаблоне с другим лицом

Теперь возьмем в качестве шаблона лицо из базы данных Olivetti faces (Рисунок 82). Проведя исследования на его основе, мы получили следующие результаты (Рисунки 83-96).



Рисунок 82. Шаблон из базы лиц Olivetti



Рисунок 83. Результат для эталонного изображения

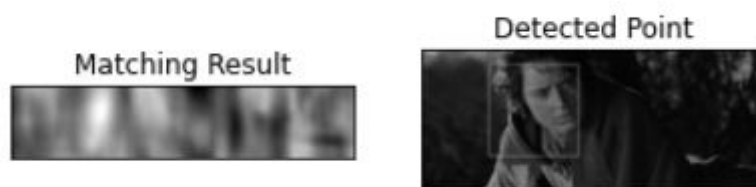


Рисунок 84. Результат для плохого освещения

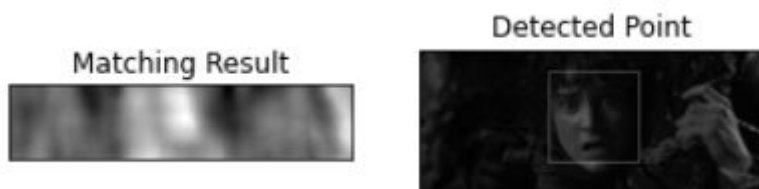


Рисунок 85. Результат для неравномерного освещения

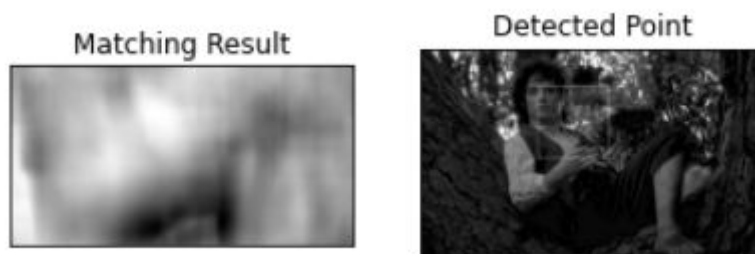


Рисунок 86. Результат для удаленности от камеры





Рисунок 87. Результат для скрытой части лица

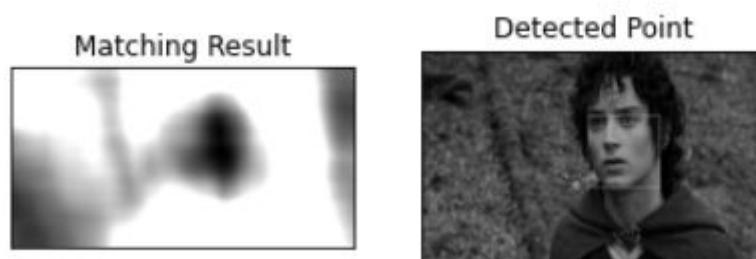


Рисунок 88. Результат для поворота головы

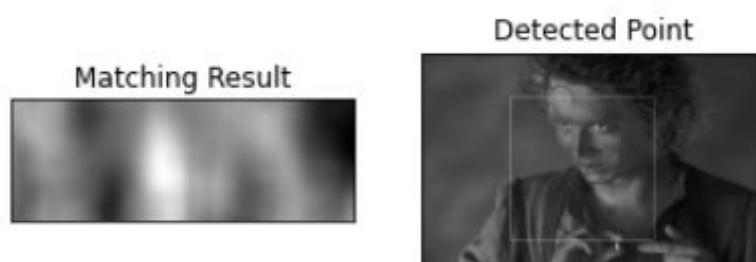


Рисунок 89. Результат для наклона головы



Рисунок 90. Результат для эталонного изображения

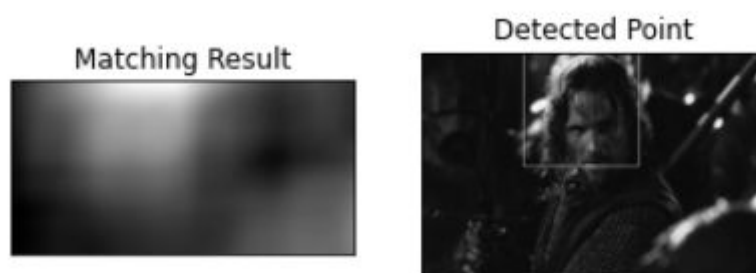


Рисунок 91. Результат для плохого освещения

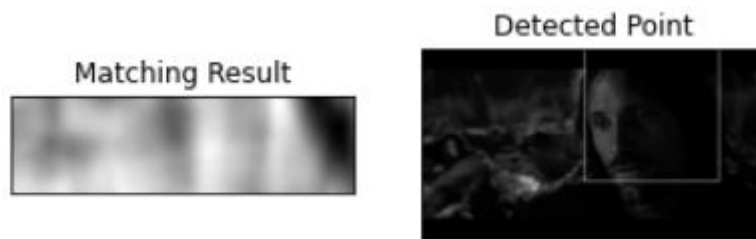


Рисунок 92. Результат для неравномерного освещения

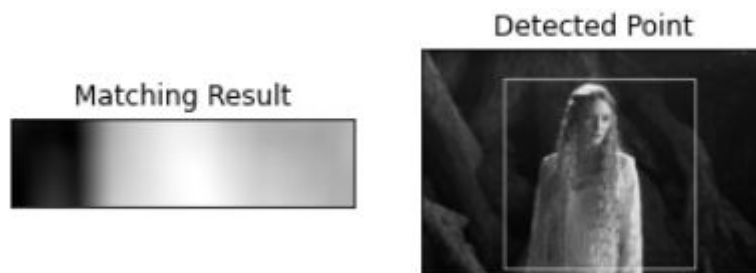


Рисунок 93. Результат для удаленности от камеры



Рисунок 94. Результат для скрытой части лица

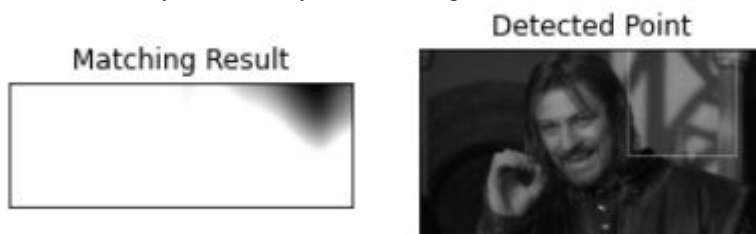


Рисунок 95. Результат для поворота головы



Рисунок 96. Результат для наклона головы

Как мы видим, метод Template Matching с другим шаблоном ведет себя достаточно адекватно, что сообщает нам о корректности его работы.

## Выводы

Все методы, описанные выше, показали себя по разному: в одних случаях лучше, чем другие, в других - наоборот. Результаты их работы были занесены в Таблицу 2, представленную ниже. В ней строки с “+” в названии означают результаты тестирования на других лицах.

	Идеальное фото	Плохое освещение	Неравномерное освещение	Удаленность от камеры	Часть лица скрыта	Поворот лица	Наклон лица
ТМ	+	+	+/-	+	+	+	+/-
ТМ+	+	+/-	+/-	+	+	+/-	+
VJ	+	-	-	+	-	+	-
VJ+	+	-	+	+	+	+	-
ЛСЛ	+	+	+	+	-	+	-
ЛСЛ+	+	-	-	+	-	+	-

Таблица 2. Результаты работы алгоритмов на различных пакетах фотографий

Как мы видим, все методы в той или иной мере справились с определением лица в идеальных условиях, условиях удаленности от камеры и поворота лица. Хуже всего алгоритмы показали себя при условии наклона головы.

Также заметно, что лучше всего справился Template Matching, однако стоит помнить, что в него входит 6 методов и что таких идеальных результатов в таблице мы достигли с помощью их комбинирования (применения оптимальных методов Template Matching при разных условиях съемки).

Стоит отметить, что в таблицу не внесен результат работы программы “Fraunhofer”, так как она не была реализована нами, однако стоит отметить, что данное приложение со своей задачей справилось прекрасно.

## Заключение

В результате проделанной работы были реализованы методы Template Matching и Viola-Jones (с простейшим интерфейсом), а также алгоритм поиска линий симметрии лица. Данные алгоритмы были протестированы на различных шаблонах и при разных условиях съемки.

Была протестирована демо версия программы определения лиц “Fraunhofer”.

В результате мы получили, что все методы в той или иной мере справились со своими задачами. Лучшего всего себя показали программа “Fraunhofer” и реализованный нами алгоритм Template Matching (с оговоркой, упомянутой в главе “Выводы”).