Сегментация водных загрязнений на линзе камеры по одному изображению на основе архитектуры U-Net

Кондрашов К.П.[1,2,*], Плохотнюк В.О.[1,2], Ершов Е.И.[1,2]

[1] Институт проблем передачи информации им. А.А. Харкевича
[2] Московский физико-технический институт (НИУ)
[*] kirpall@ya.ru

Аннотация Для избежания ошибок распознавания в системах технического зрения полезно автоматически определять, загрязнены ли их камеры, а также выделять загрязненную область. Классические методы выделения основываются на видеопоследовательностях, однако часто мы имеем доступ только к одному изображению. В работе рассмотрен метод сегментации капель, конденсата и струек воды по одному изображению сцены на основе нейросети на базе U-Net с использованием дополнительных каналов-экстракторов на основе дискретного вейвлет преобразования и насыщенности. В результате была обучена нейронная сеть, демонстрирующая высокие показатели метрик сегментации (Accuracy 0.87, Precision 0.83, Recall 0.72, IoU 0.63), а также использование экстракторов показало незначительное улучшение метрик, в частности, IoU на 1% и Recall на 1.4%.

Ключевые слова: детекция дождевых капель, дискретное вейвлет преобразование, сверточные сети, семантическая сегментация, U-Net

1 Введение

Многие системы технического зрения подвержены загрязнению камер. Особенно это актуально для камер наружнего наблюдения [14] и автономных движущихся систем (АТС, домашние роботы, дроны) [13]. Грязь или дождевые капли могут сильно ухудшить качество распознавания [15] и привести к последствиям, опасным для человека, например, скрывать от АТС препятствие или пешехода. Такие системы должны вовремя реагировать на загрязнение, запуская очистку, остановку или предупреждая владельца, и для этого полезно знать область загрязнения. При достаточно точной сегментации загрязнения, система технического зрения может продолжать свою работу, игнорируя сигнал из загрязнённой области.

1.1 Описание задачи

Внимание данного исследования будет сфокусировано на сегментации водного загрязнения **(рис. 1)** на защитной линзе камеры по одному изображении сцены. Мы будем основываться на том, что линза расположена не

вплотную к объективу, чтобы загрязнение не всегда покрывало изображение полностью, но при этом не слишком удалена (фокусное расстояние кратно больше расстояния до линзы), чтобы капли всегда оставались вне фокуса камеры. Результатом сегментации будет бинарная маска области, принадлежащей загрязнению **(рис. 2)**.

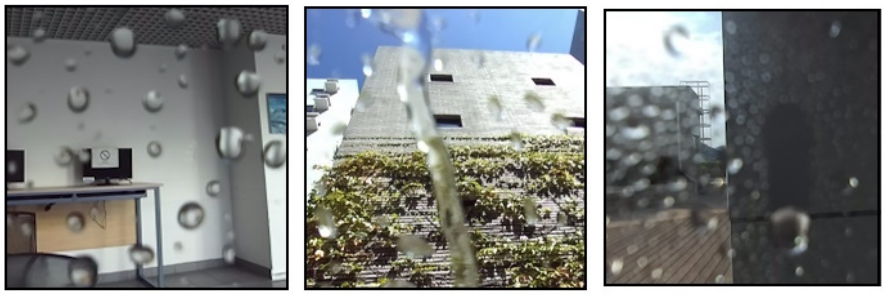

Рис. 1. Виды водного загрязнения: (a) капли (b) струйки воды (c) мелкие капли (конденсат)

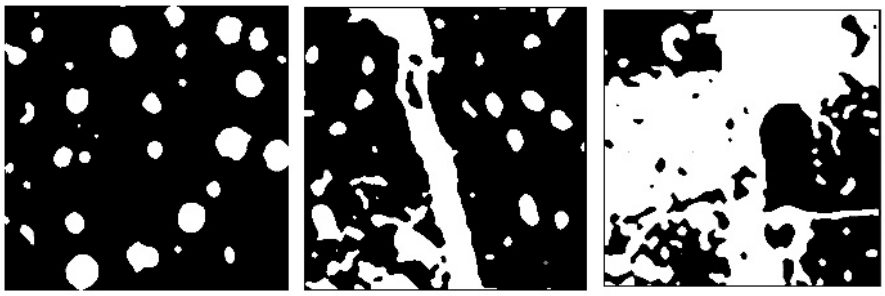

Рис. 2. Примеры бинарных масок. Белым отмечены загрязненные пиксели. Маски соответствуют рисунку 1. Заметно, что разметка наших данных неидеальна.

В нашей работе представлена комбинация нейросетевого и статистического подхода для решения данной задачи. Мы исследуем влияние дополнительных признаков, полученных статистическими методами, на качество распознавания выбранной нами нейросети, а также в принципе исследуем эффективность использования нейросети для сегментации капель.

1.2 Обзор литературы

Существует много работ, посвященных детекции водных загрязнений на линзе камеры (линзой может выступать лобовое стекло АТС). Самым распространённой задачей среди них является детекция капель по видеопоследовательности. Например, в статьях [1], [2], [3] и [4] используется последовательность кадров с камеры, закреплённой внутри или снаружи автомобиля. Методы этих статей основываются на двух предположениях:

1. капли на линзе неподвижны на всей последовательности кадров.
2. сцена находится в движении относительно камеры.

Стоит заметить, что при высоких скоростях движения и определенной форме защитного стекла возможно невыполнение первого условия из-за непрерывного движения капель. При этом второе условие не позволяет применять данные методы для задач, где сцена в основном статична. Однако на подходящих задачах эти методы дают быстрый результат с хорошей точностью, поэтому стоит рассмотреть некоторые из них. В [1] происходит максимизация градиента изображения по последовательности и выбор пикселей с наименьшим максимальным градиентом. В работе [4] подсчитывается попарная корреляция одного и того же пикселя для соседних изображений в последовательности. Утверждается, что загрязнённые пиксели будут мало меняться во времени и высоко коррелировать друг с другом. Статья [2] предлагает рассматривать изображение как двумерный сигнал и выделять границы крупных капель фильтром, основанном на дискретном вейвлет-преобразовании. Капли на линзе всегда находятся вне фокуса и их границы размыты, и на третьем уровне преобразования размытые границы хорошо видны, однако с каждым уровнем карта границ уменьшается в 4 раза, а интенсивность шумов растёт, что усложняет поиск некрупных капель. Именно поэтому авторы используют этот метод на последовательности и усредняют результат, тем самым борются с шумом.

Поскольку множество работ полагаются на видеопоследовательность, наиболее актуально исследование именно детекции по одному изображению, тем более что задачи, сводящиеся к поиску капель, не всегда подразумевают съемку видео. В статье [5] и патенте [6] представлены два метода детекции, основанные на необычном подходе: определять ROI (области интереса) изображения, а затем проверять схожесть области с искусственно смоделированной каплей, если бы она была в этом месте. Ограничения такого метода в высоких вычислительных затратах и в предположении, что капли имеют идеальную эллиптическую форму. DWT предлагали использовать не только авторы [2], но и [3] в задаче классификации загрязнения на линзе, где суммарная энергия вейвлетов в разложении служила одним из признаков, размыта область или нет. Поскольку капли представляют собой размытые области, этот признак представляет для нас интерес. Нейросетевые методы представлены в основном очисткой изображения от капель и конденсата, и используют GAN [7] или Pix2PixHD архитектуру с Residual блоками [8]. Оба метода показывают высокую точность на своей задаче. Кроме этого,

в [8] применяли архитектуру U-Net для сегментации дорожной разметки на загрязненных каплями изображениях, и также получили неплохие показатели, что указывает на возможность применения архитектуры U-Net и для нашей задачи.

Исходя из сказанного в предыдущем пункте мы выдвинули две гипотезы:

1. Данную задачу возможно успешно решать, используя нейронную сеть архитектуры U-Net. Успешность определяется метриками сегментации.
2. Мы можем улучшить качество распознавания, используя дискретное вейвлет-преобразование как источник дополнительной информации для нейросети.

2 Предложенный метод

Нам требуется нейронная сеть, способная обучиться на небольшом числе изображений. Нашим выбором стал U-Net — относительно быстрая масштабируемая архитектура, подходящая для задачи семантической сегментации. Её структура изображена на рисунке **6**.

Внутрь сети встраиваются дополнительные каналы — каналы-экстракторы — с целью улучшения качества сети.

2.1 Каналы экстракторы

Для улучшения качества мы можем сами выделить некоторые признаки капель и подать их на вход нейросети вместе с изображением в качестве дополнительных каналов. Рассмотрим подходящие признаки и полученные из них каналы, названные экстракторами.

Дискретное вейвлет-преобразование Как было сказано выше, двумерное DWT используется для выделения размытых границ. Для превращения DWT в канал для нейросети, мы сложим частотные компоненты $LL_1, LH_1, HL_1, HH_1, \ldots$ [12] в мозайку **(рис. 3)**, нетрудно заметить, что она будет иметь тот же размер, что и подаваемое в сеть изображение. Разложение ведётся до третьего уровня, ровно как в [2] и [3].

Однако этот метод возможно улучшить, передавая не один канал, а все частотные компоненты первых трёх уровней. Каждый новый уровень возвращает компоненты в 4 раза меньшие по размеру, чем предыдущие, ровно как и U-Net на каждой ступени свёртки уменьшает feature map в 4 раза. Таким образом, мы имеем соответствие между уровнями DWT и ступенями свёртки U-Net, а значит, можем подать $i-$ю частотную компоненту в $i+1-$ю ступень, как показано на рисунке **6**.

Насыщенность Было обнаружено, что водные загрязнения могут уменьшать насыщенность фона, что позволяет нам выделять их по каналу насыщенности изображения**(рис. 4)**. В случае, когда в сеть подаются патчи с цветовой моделью RGB, можно попробовать облегчить ему задачу этим каналом.

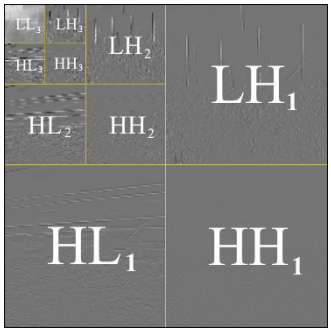

Рис. 3. Канал DWT с разметкой компонент. Показано, как именно составляется мозайка.

Рис. 4. Канал насыщенности. Тёмные пятна — это капли.

2.2 Алгоритм сегментации

Здесь мы опишем алгоритм сегментации с учётом того, какое множество каналов-эстракторов мы подсчитываем для нейросети. От этого зависит скорость и качество используемого нами метода.

Алгоритм

Входные данные: трёхканальное изображение в цветовой модели RGB или HSV, список используемых каналов-экстракторов, пороговое значение τ

1. Посчитать для изображения каналы-экстракторы.
2. Подать изображение и каналы на вход U-Net.
3. Получить предсказание и произвести бинаризацию по τ.

Выходные данные: Полученная бинарная маска.

Значение τ выбирается вручную на основе наилучшего визуального соответствия. На наших данных был выбран $\tau = 0.3$. На рисунке **5** представлена схема вычислений.

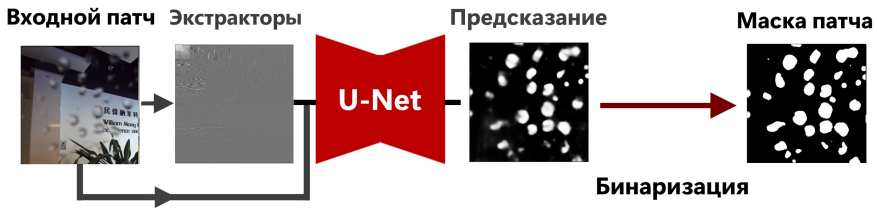

Рис. 5. Схема вычислений. U-Net получает входной патч и экстракторы. Экстракторы считаются из входного патча.

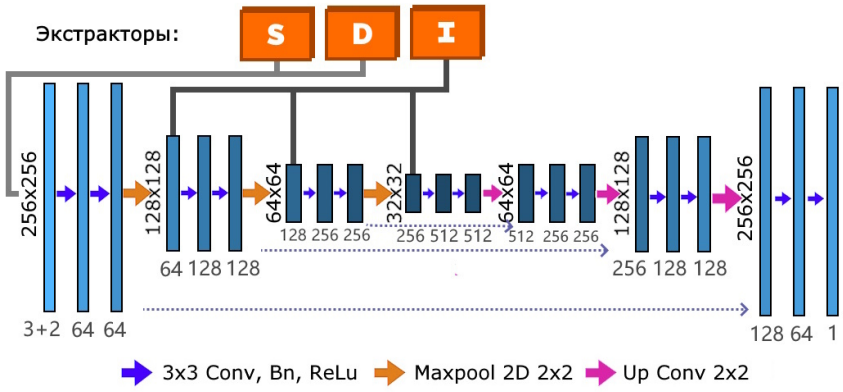

Рис. 6. Структура использованной нами сети U-Net, а также места вставки каналов-экстракторов. 'S' — экстрактор насыщенности, 'D' — DWT мозайка, 'I' — компоненты DWT трёх уровней. На вторую ступень U-Net передаются LH_1, HL_1, HH_1, на третью LH_2, HL_2, HH_2 и на четвертую LH_3, HL_3, HH_3

3 Постановка эксперимента

Для оценки эффективности мы обучим несколько нейросетей, подавая различные множества экстракторов. Для обучения нам потребуются наборы данных, содержащие упомянутые нами загрязнения **(рис. 1)** и соответствующие бинарные маски **(рис. 2)**. Однако мы не нашли уже размеченных наборов с каплями на линзе, поэтому встала задача сделать разметку самостоятельно.

3.1 Разметка данных

Для задачи очистки изображения от капель собраны несколько наборов данных [8,9], содержащих пары чистое/загрязненное изображение, из чего мы можем разметить маски сами. Мы выбрали 2 из них:

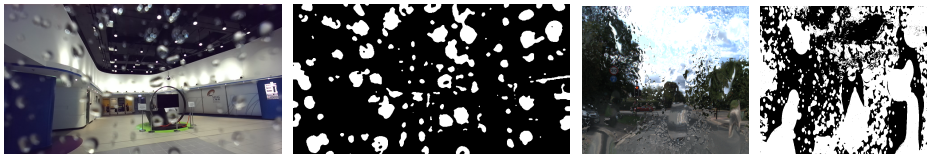

Рис. 7. Пример изображения и его маски для «Стерео» набора данных (слева) и CG (справа). Маски размечены предложенным ниже методом.

1. Набор данных «Стерео» из [9] содержит все три вида загрязнения. Он состоит из групп изображений по 6 стерео пар снимков одной локации. Первая пара — чистые изображения. Оставшиеся содержат капли с разной степенью загрязненности. Пары снимаются без смещения штатива, но не одновременно, что приводит к разнице освещения между чистыми и грязными снимками в группе. Основные локации — интерьеры (71%), пешеходные улицы (29%). Капли различных форм и размеров.
2. Набор данных «CG» из [8] содержит компьютерно сгенерированные капли, пары чистое/грязное. Большая плотность покрытия **(рис. 8)**, разнообразный размер капель. Основан на видео автомобильной езды по Оксфорду.

Для CG разметка очевидна, поскольку грязная картинка попиксельно эквивалентна чистой в областях, не затронутых CG-каплями. Метод поиска маски: найти попиксельную разницу между чистым и грязным изображением и выбрать пиксели с заметной разницей бинаризацией по некоторому $\tau > 0$.

$$M_c(x,y) = \begin{cases} 1, & \text{if } |A_c(x,y) - B_c(x,y)| \geq \tau \\ 0, & \text{if } |A_c(x,y) - B_c(x,y)| < \tau \end{cases}$$

где A_c — c-тый канал чистого изображения A, B_c — c-тый канал грязного изображения B, M_c — бинарная маска по c-му каналу. Будем считать, что пиксель попал в итоговую маску, если хотя бы один его канал c попал в M_c. Тогда итоговая маска:

$$M(x,y) = \max_{c \,\epsilon\, R,G,B} M_c(x,y)$$

где M — итоговая маска изображения B.

Этим методом мы разметили «Стерео» набор, выбрав $\tau = 0.3 \times 256 = 76,8$. Однако возникла проблема: за время между съемкой пары зачастую изменяется освещение, что вносит существенную попиксельную разницу. Например, при изменении угла солнца, сместившиеся тени создают сильно измененные полосы и пятна. То же касается и движения облаков.

Для исправления были вручную удалены все картинки, где ошибки занимали большу́ю площадь. Итого для обучения оставлено 791 изображение «Стерео» набора и 4816 CG-набора. Оба набора были разбиты на тренировочную (80%), валидационную (10%) и тестовую (10%) выборки.

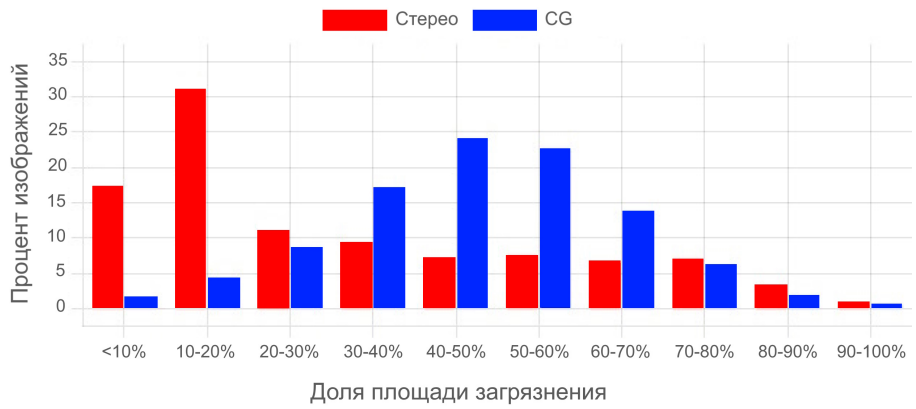

Рис. 8. Распределение плотности загрязнения в наборах данных. «Стерео» содержит в основном разреженные загрязнения, в CG распределение близко к нормальному.

3.2 Обучение

Сеть обучается на случайных нормализованных патчах 256×256 пикселей с использованием функции потерь **BCEWithLogitsLoss** [10] и оптимизатора **Adam** [11]. Фреймворк для обучения — **PyTorch 2.0.1+cu117**. Обучение производится на каждом из двух наборов данных по отдельности.

Для «Стерео» набора изначальный $learing\ rate = 0.004$, был подобран sheduler **MultiStepLR** с параметрами $milestones = [10, 30, 50, 70, 85], gamma = 0.63$ в течение 150 эпох до переобучения. Размер батча 16.

Для «CG» изначальный $learing\ rate = 0.004$, sheduler **MultiStepLR** с параметрами $milestones = [5, 7, 15, 25], gamma = 0.63$ в течение 40 эпох. Размер батча 32.

Код обучения и подготовки наборов данных выложен в репозитории Github [16].

3.3 Метрики

Введём простейшие метрики оценки сегментнации: Accuracy, Precision, Recall, IoU (Индекс Джаккарда).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \quad Precision = \frac{TP}{TP + TN},$$

$$Recall = \frac{TP}{TP + FP}, \quad IoU = \frac{TP}{TP + FN + FP},$$

где TP — число верно предсказанных загрязненных пикселей, TN — верно предсказанных чистых пикселей, FP — неверно предсказанных чистых пикселей, FN — верно предсказанных загрязненных пикселей.

4 Результаты

В этой секции мы представляем результаты, полученные с помощью нашего метода на двух наборах данных, экспериментируя с цветовой моделью патчей и набором экстракторов.

Для таблицы обозначим наши каналы:

- 'R', 'G', 'B' — красный, синий и зелёный каналы в 'RGB' соответственно.
- 'H', 'S', 'V' — аналогично с 'HSV'.
- 'D' — канал DWT.

Набор использованных каналов описывается последовательностью символов. Каждый эксперимент характеризуется своим набором каналов. Для каждого эксперимента было обучено от 10 до 35 нейросетей (точное количество указано в таблице в репозитории [16]), арифметическое среднее результатов и их стандартное отклонение были записаны в соответствующие строки таблиц результатов и отклонений. Таблицу со всеми результатами и логами запусков можно найти в репозитории Github [16].

Таблица 1. Результаты обучения на «Стерео»

Каналы	Train BCE loss	Test BCE loss	Accuracy	Precision	Recall	IoU
RGB	**0.185**	0.313	0.868	0.828	0.724	0.626
HSV	0.194	0.326	0.86	0.829	0.706	0.614
RGBS	0.192	**0.298**	**0.871**	0.839	0.715	0.626
RGBD	0.187	0.31	0.867	**0.84**	0.717	0.628
RGBI	0.186	0.311	**0.871**	0.827	**0.734**	**0.632**

Таблица 2. Результаты обучения на «CG»

Каналы	Train BCE loss	Test BCE loss	Accuracy	Precision	Recall	IoU
RGB	0.088	0.099	0.955	0.97	0.927	0.903
HSV	0.097	0.103	0.952	0.973	0.918	0.896
RGBS	0.09	0.099	**0.957**	0.971	**0.928**	0.904
RGBD	**0.087**	**0.094**	**0.957**	0.972	**0.928**	**0.906**
RGBI	0.088	0.103	0.949	**0.975**	0.914	0.895

Мы получили высокие показатели IoU для обоих наборов данных, что показывает эффективность применения U-Net к нашей задаче.

Цветовая схема HSV оказалась хуже RGB почти по всем показателям и на «CG» и на «Стерео».

Использование экстракторов 'D' и 'I' не принесло однозначных результатов: относительно лучшего результата без экстракторов на реальных данных экстрактор 'I' продемонстрировал улучшение IoU на 1%, Accuracy на 0.35%, Recall на 1.4%, а экстрактор 'D' улучшил Precision на 1.5%, однако эти результаты не являются статистически значимыми, поскольку соответствующие показатели стандартного отклонения колеблются от 1.6% до 3.8%, что превышает наши улучшения, иногда кратно.

На синтетических данных результаты 'I' не выглядят репрезентативными ввиду слишком высоких значений стандартного отклонения (в обучении встречались выбросы), однако в общем результаты оказались ниже RGB. А вот экстрактор 'D' продемонстрировал улучшение всех метрик, хотя эти улучшения так же не превысили отклонений.

Экстрактор 'S' оказал необычное действие на результат. На реальных данных Recall оказался хуже модели без экстракторов. Однако остальные метрики для обоих наборов не ухудшились, при этом стандартное отклонение Accuracy, Recall и IoU оказалось ниже конкурентов в среднем на 43% и 56%, т.е. результаты обучения с этим экстрактором сильно стабильнее.

Стоит отметить, что на обоих наборах данных экстракторы смогли повысить все имеющиеся метрики хотя бы на 0.001, что всё же доказывает их положительное влияние.

Таблица 3. Стандартное отклонение результатов на «Стерео»

Каналы	Train BCE loss	Test BCE loss	Accuracy	Precision	Recall	IoU
RGB	**0.007**	0.044	0.016	0.03	0.029	0.018
HSV	0.009	0.038	0.014	**0.023**	0.033	0.024
RGBS	0.014	**0.017**	**0.008**	0.026	**0.015**	**0.011**
RGBD	0.008	0.034	0.013	0.028	0.03	0.019
RGBI	0.008	0.049	0.014	0.024	0.027	0.017

Таблица 4. Стандартное отклонение результатов на «CG»

Каналы	Train BCE loss	Test BCE loss	Accuracy	Precision	Recall	IoU
RGB	0.004	0.019	0.016	0.01	0.026	0.021
HSV	0.004	**0.012**	0.009	**0.006**	0.021	0.018
RGBS	0.004	0.014	**0.007**	0.011	**0.02**	**0.012**
RGBD	**0.003**	0.021	0.02	0.008	0.032	0.028
RGBI	**0.003**	0.044	0.175	0.178	0.17	0.167

Наконец, мы считаем основными результатами статьи те, что получены на «Стерео» данных, а не CG, поскольку именно реальные, а не синтетические капли будут встречаться при применении нашего метода. Именно эти результаты мы вынесли в аннотацию.

5 Заключение

В данной работе мы предложили метод сегментации капель и иных водных загрязнений на линзе камеры по одному изображению. Из двух поставленных во введении гипотез подтвердилась лишь одна. Архитектура нейронной сети U-Net действительно является эффективным методом сегментации водных загрязнений по одному изображению, однако наш подход к её улучшению, использующий дискретное вейвлет преобразование, принес лишь незначительное увеличение точности по нескольким метрикам.

Список литературы

1. Hsien-Chou Liao, De-Yu Wang, Ching-Lin Yang and Jungpil Shin, 2013. Video-based Water Drop Detection and Removal Method for a Moving Vehicle. Information Technology Journal, 12: 569-583.
2. Zhang, Yi & Yang, Jie & Liu, Kun & Zhang, Xiang. (2008). Self-detection of optical contamination or occlusion in vehicle vision systems. Optical Engineering - OPT ENG. 47. 10.1117/1.2947578.
3. V. Akkala, P. Parikh, B. S. Mahesh, A. S. Deshmukh and S. Medasani, "Lens adhering contaminant detection using spatio-temporal blur,"2016 International Conference on Signal Processing and Communications (SPCOM), Bangalore, India, 2016, pp. 1-5, doi: 10.1109/SPCOM.2016.7746664.
4. Einecke, Nils & Gandhi, Harsh & Deigmoller, Jorg. (2014). Detection of camera artifacts from camera images. 2014 17th IEEE International Conference on Intelligent Transportation Systems, ITSC 2014. 603-610. 10.1109/ITSC.2014.6957756.
5. Roser, Martin & Geiger, Andreas. (2009). Video-based raindrop detection for improved image registration. 570 - 577. 10.1109/ICCVW.2009.5457650.
6. Lawson, C., Gao, L., Gagnon, P., Yadav, D. (2017) DETECTION OF WATER DROPLETS ON A VEHICLE CAMERA LENS. European patent Office. No. EP3144853A1
7. R. Qian, R. T. Tan, W. Yang, J. Su, and J. Liu, “Attentive generative adversarial network for raindrop removal from a single image,” CoRR, vol. abs/1711.10098, 2017
8. Porav, Horia & Bruls, Tom & Newman, Paul. (2019). I Can See Clearly Now: Image Restoration via De-Raining. 7087-7093. 10.1109/ICRA.2019.8793486.
9. Zifan Shi, Na Fan, Dit-Yan Yeung, Qifeng Chen. Stereo Waterdrop Removal with Row-wise Dilated Attention (2021) arXiv:2108.03457
10. https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html
11. https://pytorch.org/docs/stable/generated/torch.optim.Adam.html
12. Vu, Phong V., and Damon M. Chandler. “A fast wavelet-based algorithm for global and local image sharpness estimation.” Signal Processing Letters, IEEE 19.7 (2012): 423-426.
13. A. Das et al., "TiledSoilingNet: Tile-level Soiling Detection on Automotive Surround-view Cameras Using Coverage Metric,"2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 2020, pp. 1-6, doi: 10.1109/ITSC45102.2020.9294677.

14. Gu, Jinwei & Ramamoorthi, Ravi & Belhumeur, Peter & Nayar, Shree. (2009). Removing Image Artifacts Due to Dirty Camera Lenses and Thin Occluders. ACM Trans. Graph.. 28. 10.1145/1661412.1618490.
15. J. Li, Z. Li, X. Xu and G. Jing, "A method on Face Recognition of Contaminated Small Sample,"in 2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Houston, TX, USA, 2021 pp. 3392-3399. doi: 10.1109/BIBM52615.2021.9669558
16. https://github.com/KIrillPal/WaterDropDetection