



УНИВЕРСИТЕТ ИТМО

VGLib: Библиотека по обработке и генерации векторной графики

Жарский Иван и команда VGLib

Обо мне

Исследователь в лаборатории КТ ИТМО в команде по обработке и генерации векторной графики. Соавтор в пяти статьях по теме векторной графики.

Техлид и CV-инженер в компании Statanly

Аспирант 1 курса ФИТиП ИТМО

Веду практики по машинному обучению

Занимаюсь векторной графикой с конца 2021 года



Иван Жарский
TG: @IzhanVarsky

Содержание:

1. Векторное изображение – что это?
2. Методы обработки векторной графики, доступные в VGLib
3. Планы на будущее

Векторное изображение – что это?

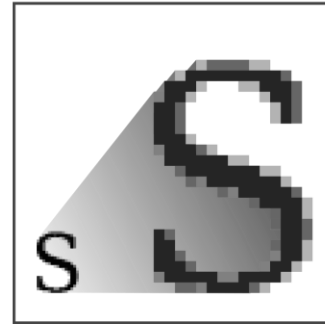
Растровое изображение

Матрица пикселей



Векторное изображение

- Будем рисовать последовательно геометрические фигуры на холсте
- Каждая фигура задается своими уравнениями и параметрами
- Каждой фигуре можно указать её цветовые характеристики



РАСТР
.jpeg .gif .png

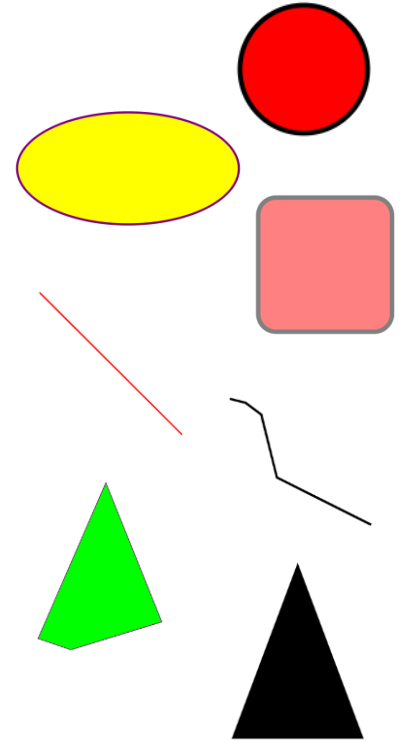


ВЕКТОР
.svg

$$\text{Окружность: } (x - x_0)^2 + (y - y_0)^2 = R^2$$

Базовые SVG фигуры

Тег	Атрибуты
<code><circle></code>	<code>cx="50" cy="50" r="40"</code>
<code><ellipse></code>	<code>cx="200" cy="80" rx="100" ry="50"</code>
<code><rect></code>	<code>x="50" y="20" width="150" height="150" rx="20" ry="20"</code>
<code><line></code>	<code>x1="0" y1="0" x2="200" y2="200"</code>
<code><polyline></code>	<code>points="20,20 40,25 60,40 80,120 120,140 200,180"</code>
<code><polygon></code>	<code>points="220,10 300,210 170,250 123,234"</code>
<code><path></code>	<code>d="M150 0 L75 200 L225 200 Z"</code>



Тег <path> - кубическая кривая

```
<svg width="480" height="240" viewBox="0 0 1200 600">
```

```
<path d="M200,300 C 200,200 450,50 600,300 S 500,500 400,400 Z"  
      fill="none" stroke="red" stroke-width="5" />
```

```
<circle cx="200" cy="300" r="10" fill="red"/>
```

```
<circle cx="200" cy="200" r="10" fill="green"/>
```

```
<circle cx="450" cy="50" r="10" fill="green"/>
```

```
<circle cx="600" cy="300" r="10" fill="blue"/>
```

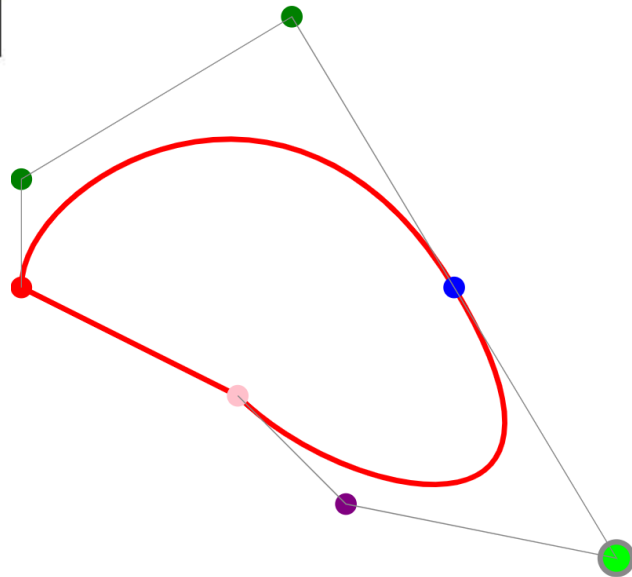
```
<circle cx="750" cy="550" r="15"  
      fill="lime" stroke="#888888" stroke-width="5"/>
```

```
<circle cx="500" cy="500" r="10" fill="purple"/>
```

```
<circle cx="400" cy="400" r="10" fill="pink"/>
```

```
<path d="M200,300 L200,200 L450,50 L600,300  
      L750,550 L500,500 400,400"  
      fill="none" stroke="#888888" stroke-width="1" />
```

```
</svg>
```

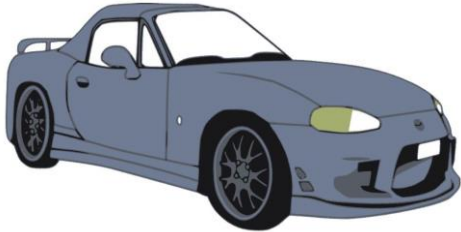


Другие теги

Tag	Описание
<g>	Группирует теги, задает общие атрибуты
<defs>	Позволяет задавать графические объекты для последующего использования
<linearGradient>	Задает линейный градиент. На тег ссылаются из атрибута fill="url(#id_name)". Дочерние теги <stop> задают используемые цвета и их границы.
<radialGradient>	То же самое, но градиент радиальный (круговой)
<style>	Такой же смысл как и в HTML
<text>	Текст

Сравнение растровых и векторных изображений

Критерий	Растровая графика	Векторная графика
Тип файла	JPG, JPEG, PNG и др.	SVG, EPS и др.
Внутреннее представление	RGB(A) матрица из пикселей	Последовательность геометрических фигур
Увеличение масштаба картинки	Явная видимость пикселей	Неограниченная масштабируемость без потери качества
Объем занимаемой памяти	Не зависит от сложности объектов, но зависит от разрешения	Простые фигуры: малый размер файла. Сложные и зашумленные фигуры – объем резко возрастает
Назначение	Любые изображения, фотографии в том числе	Графический дизайн, иконки, логотипы, текстовые шрифты, иллюстрации и прочее
Генерация в машинном обучении	Отлично изучена. Существует множество моделей для классификации и генерации растровых изображений.	Изучена плохо. Основные трудности в выборе количества генерируемых фигур и их сложности.



Векторная графика



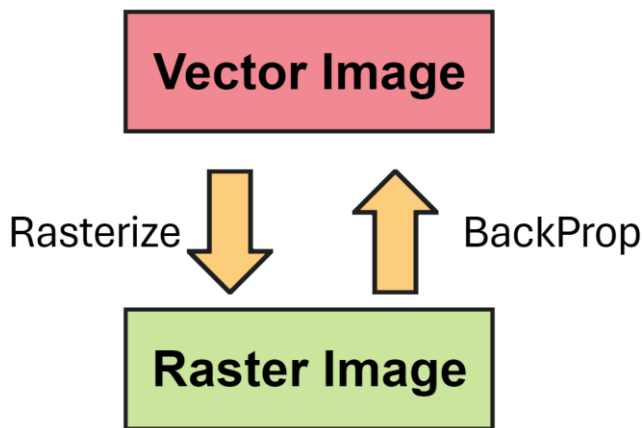
Растровая графика

Методы обработки векторных изображений, доступные в VGLib

1. CoverGAN – генератор векторных музыкальных обложек
2. VectorNST – итеративный метод переноса стиля в векторной графике
3. EvoVec – эволюционный метод векторизации изображений
4. Алгоритм проверки плагиатов чертежей
5. VectorWeaver – безусловный генератор векторных изображений

<https://github.com/CTLab-ITMO/VGLib>

DiffVG (2020)



Дифференцируемый растеризатор векторной графики

Позволяет «связать» векторную и растровую графику через обратное распространение ошибки.

Можно сконструировать векторное изображение, растеризовать его и эту растеризацию использовать в функции потери.

CoverGAN – генератор векторных музыкальных обложек

CoverGAN. Требования к сервису

Входные параметры:

- Трек музыкальной композиции.
- Имя исполнителя и композиции.
- Вызываемая эмоция.

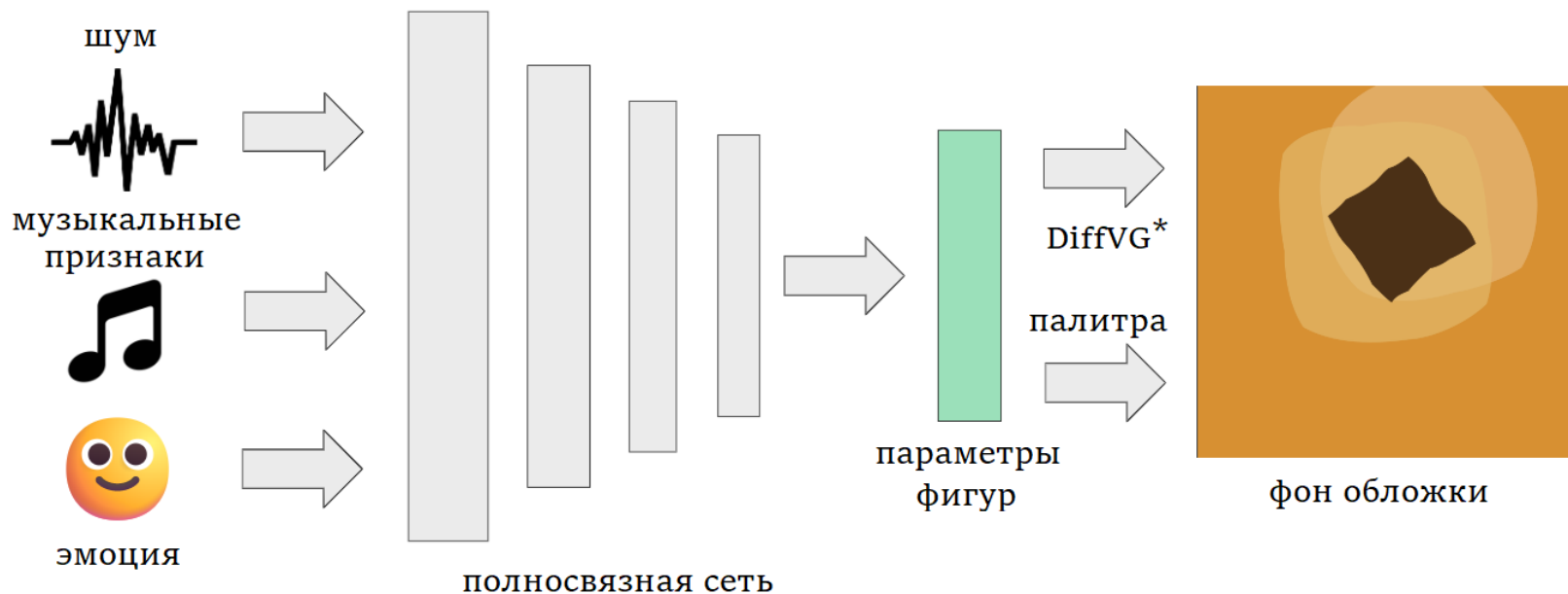
Параметры выходной обложки:

- Формат SVG.
- Подпись музыкальной композиции на обложке.



<https://www.scitepress.org/Papers/2024/124561/124561.pdf>

Архитектура генератора CoverGAN



* DiffVG – дифференцируемый растеризатор векторной графики

Примеры обложек CoverGAN



VectorNST – метод переноса стиля в векторной графике

VectorNST

Перенос стиля в векторной графике



Контент



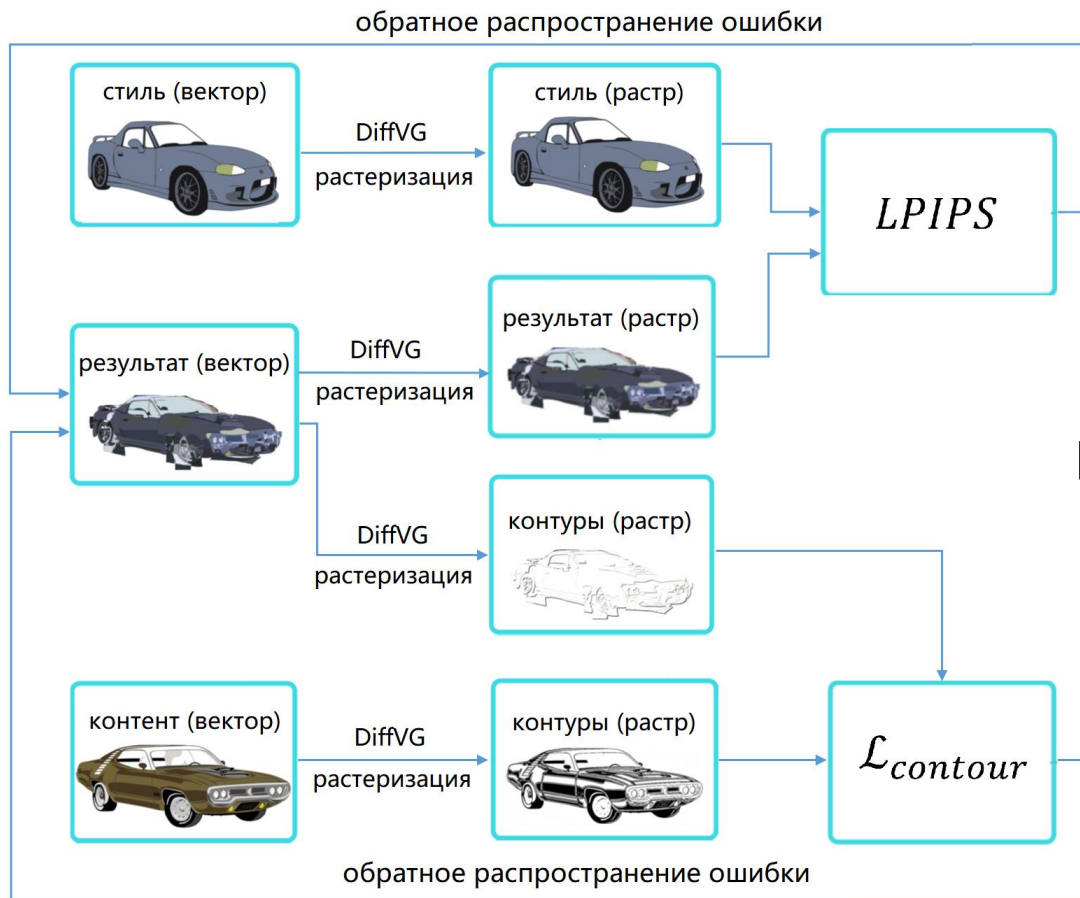
Стиль



Результат

<https://www.scitepress.org/Papers/2024/124382/124382.pdf>

VectorNST



Итеративный подход

VectorNST. Примеры работы метода

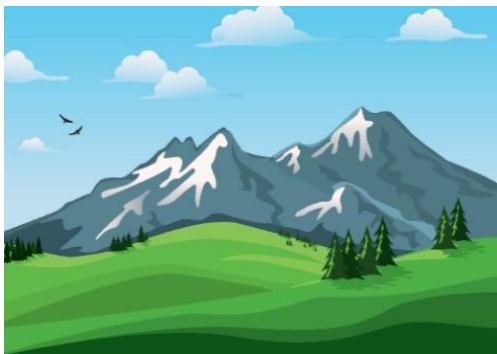
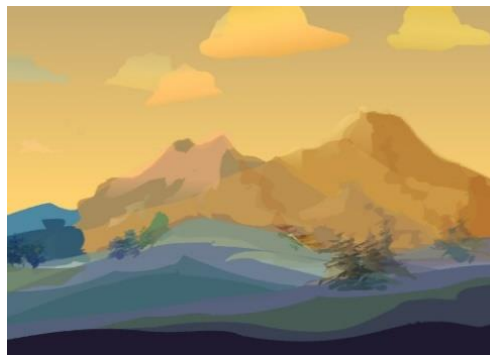
Контент



Стиль



Результат



EvoVec – метод векторизации растровых изображений






EvoVec

Векторизация изображений с помощью эволюционных алгоритмов

Алгоритм:

1. Инициализируем изображение с помощью SVGTracer
2. Эволюционным алгоритмом удаляем лишние фигуры, сегменты кривых, исправляем контуры фигур

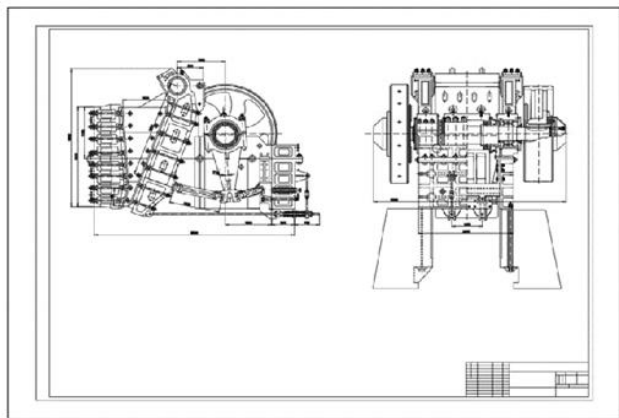
EvoVec

Исходное изображение	LIVE (N=64)	DiffVG (N=512)	SvgTracer	EvoVec	Метрика
					Результат
	51,5 м.	45,4 м.	7,5 с.	24,5 м.	Время работы
	4760	4255	1780	1547	Значение ф. отбора
	64	512	1790	985	Кол-во путей

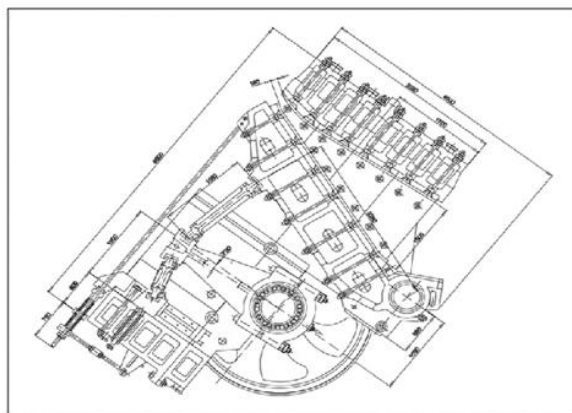
https://link.springer.com/chapter/10.1007/978-3-031-70085-9_24

Метод проверки плагиатов чертежей

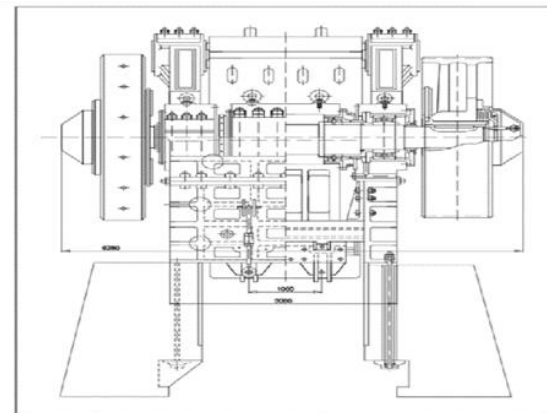
Проверка плагиатов чертежей на основе рекурсивного трансформера с памятью



Оригинальное изображение



Плагиат. Расстояние: 0.0280
Обнаружен: True



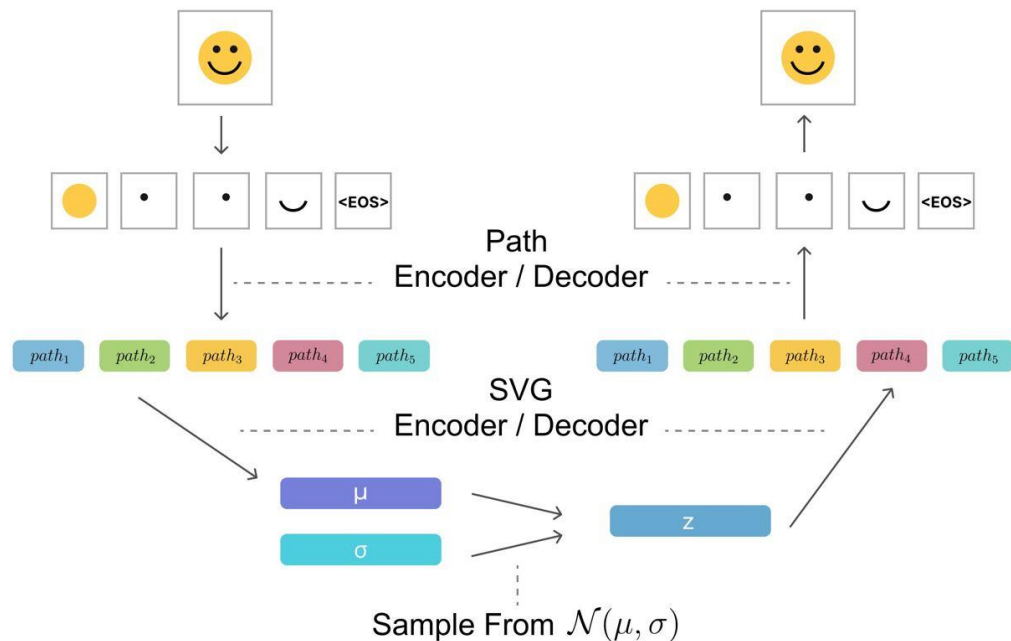
Плагиат. Расстояние: 0.0197
Обнаружен: True

Используем Bert для извлечения признаков. Обучали TripletLoss'ом

Vector Weaver – метод безусловной генерации векторных изображений

Vector Weaver

Безусловный генератор векторных изображений



Архитектура: двухэтапный VAE-трансформер



Сгенерированные изображения

Планы на будущее

Генерация с помощью ChatGPT

Запрос:

Using the SVG format, output a schematic reproduction of Malevich.

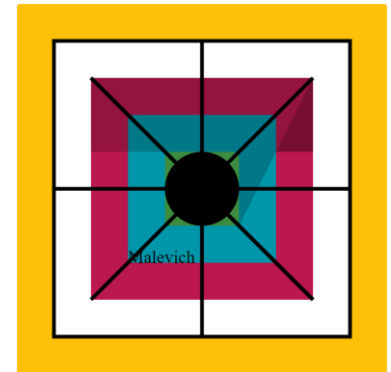
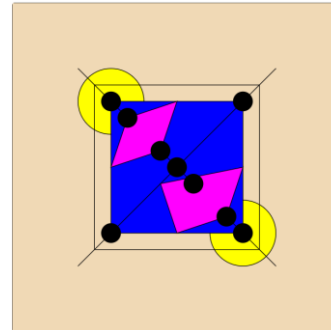
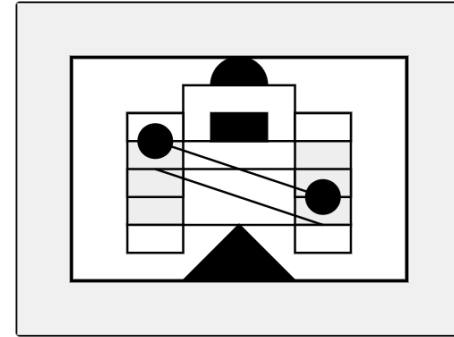
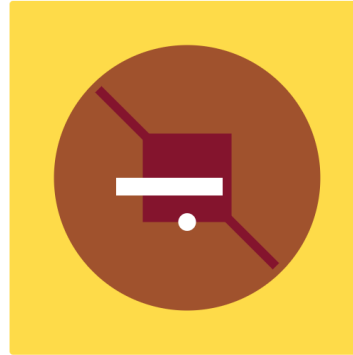
[I want to see a frame, objects, Malevich-like details].

Put the output in the code block. Use vibrant colors.

Don't forget

`xmlns="http://www.w3.org/2000/svg"` after `svg` tag.

Use 20 lines of code.



<https://neural.love/blog/chatgpt-svg>

1. Дообучить LLM на задаче кодогенерации SVG изображений
2. Обучить полноценный эмбеддер векторных фигур и изображений на основе нескольких доменов (SVG-код, растеризованное изображение, текстовое описание)
3. Обучить LLM для генерации SVG по текстовому описанию
4. Улучшить методы сопоставления чертежей
5. Реализовать метод генерации чертежей по заданному условию
6. Реализовать новые функции потерь для сглаживания неровностей краев фигур

7. Фундаментально: автоматически определять и генерировать нужное число фигур подходящей сложности
8. Фундаментальная проблема: векторные изображения – это по большей части числа. А с числами и их вычислениями LLM справляется плохо
9. VGLib: выделить общие функции пре- и пост- процессингов, генерации и аугментации фигур и изображений, лоссов
10. VGLib: расширять библиотеку методами, предложенными другими авторами

Команда VGLib

- Валерия Ефимова – руководитель
- Андрей Фильченков – руководитель
- Иван Жарский
- Егор Баженов
- Андрей Пименов
- Артем Чебыкин
- Артем Трещев
- Борис Малашенко
- и другие ...

Ссылка на библиотеку: <https://github.com/CTLab-ITMO/VGLib>

Спасибо за внимание!

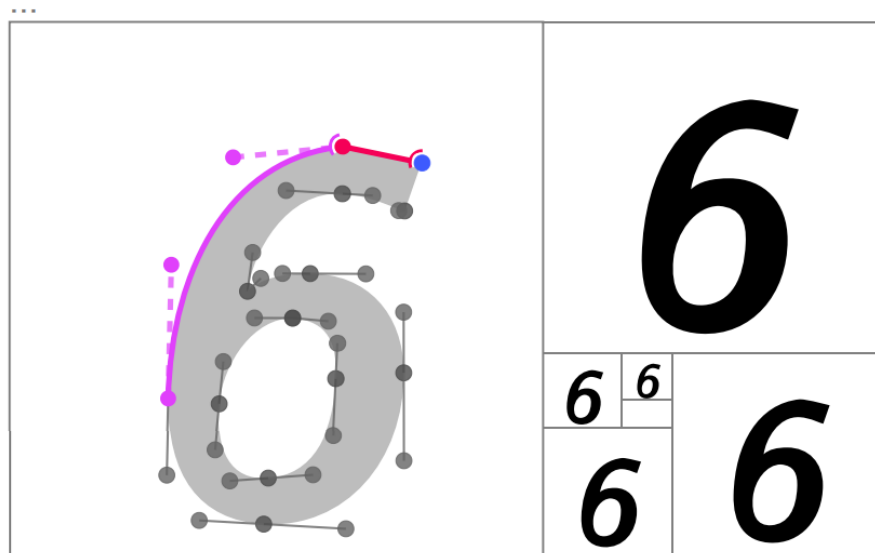
www.ifmo.ru

ITMO^s *re than a*
UNIVERSITY

Обзор ранее предложенных методов обработки векторных изображений

SVG-VAE (2019)

moveTo (15, 25)
lineTo (-2, 0.3)
cubicBezier (-7.4, 0.2) (-14.5, 11.7), (-12.1, 23.4)



Генерация шрифтов (0-9, a-z, A-Z)
по заданному стилю символа

Датасет: 14M символов.

4 типа команд:

- moveTo
- lineTo
- cubicBezier
- EOS

Генерируем один символ = одну последовательность.

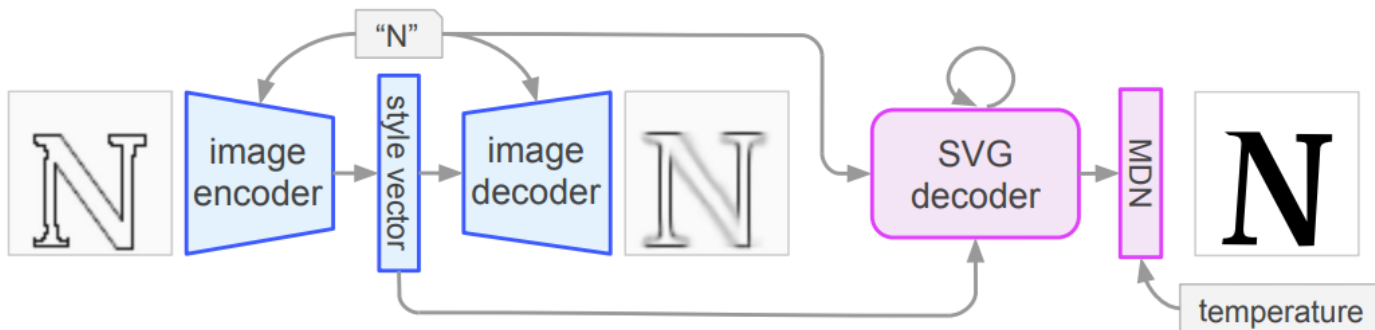
Суммарно команд не более 50.

<https://arxiv.org/pdf/1904.02632.pdf>

SVG-VAE (2019)

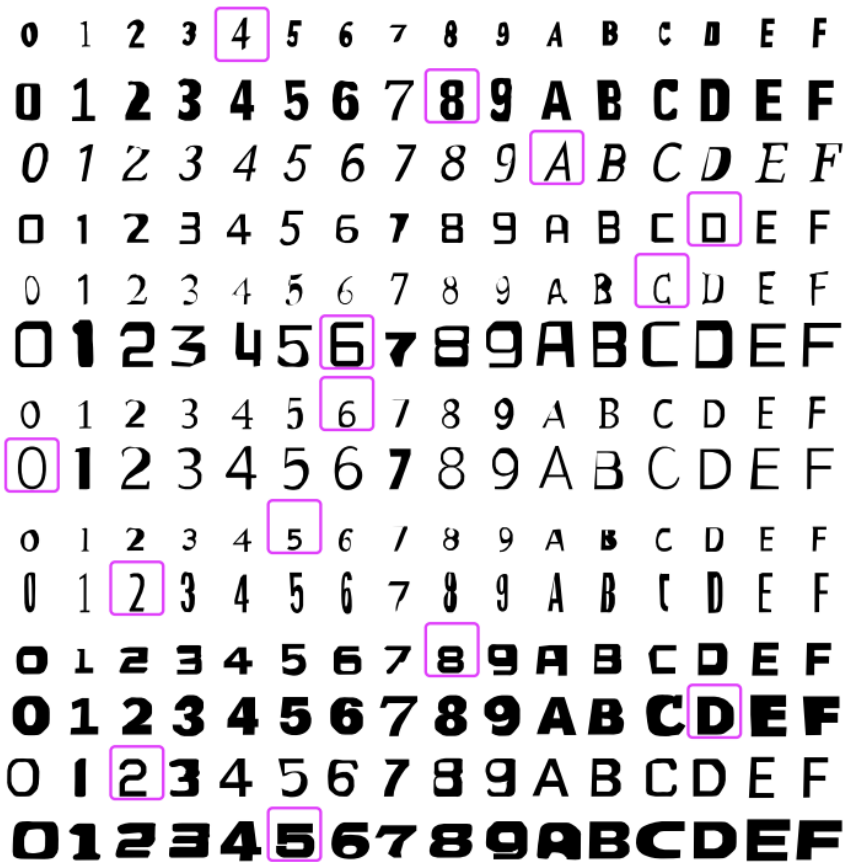
Image Autoencoder

SVG Decoder



Архитектура: Conditional VAE.
Скрытый слой – стиль шрифта

SVG-VAE (2019)



Проверка стилистики скрытого слоя:

- Выбрали вектор стиля у конкретного символа (фиолетовый квадрат)
- По этому вектору декодер сгенерировал оставшиеся символы
- Итог: стиль *почти* однообразен.

Текущие проблемы

Хотим уметь генерировать более сложные SVG:

- Несколько последовательностей (путей)
- Добавить цветогенерацию
- Учитывать прозрачность (opacity) и т.п.

Хотим использовать трансформеры

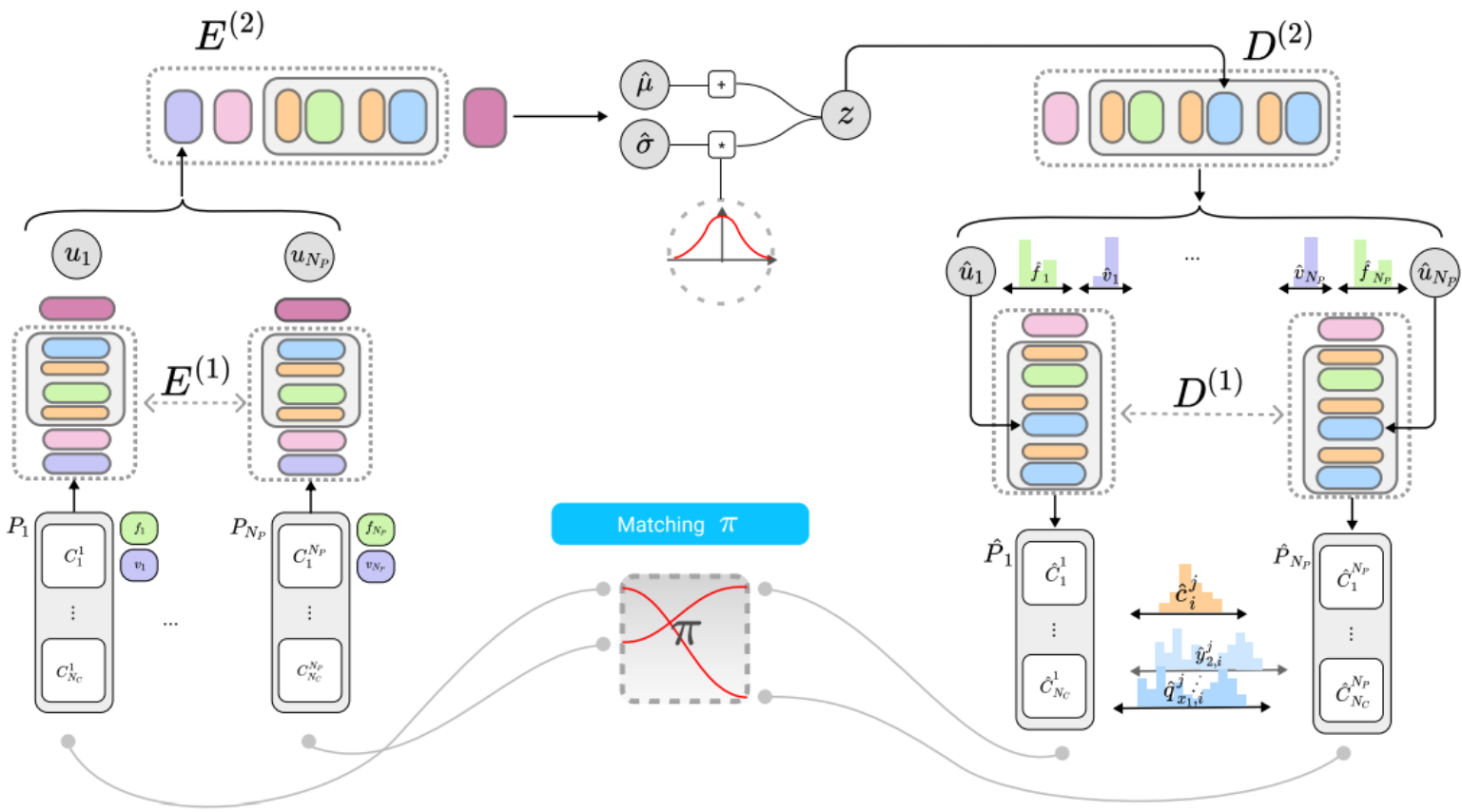
DeepSVG (2020) - идея

Идея – двухэтапный VAE:

- закодируем каждый путь, получим набор внутренних представлений
- объединим эти представления и опять закодируем
- полученное – внутреннее представление всей картинки
- аналогично дважды декодируем

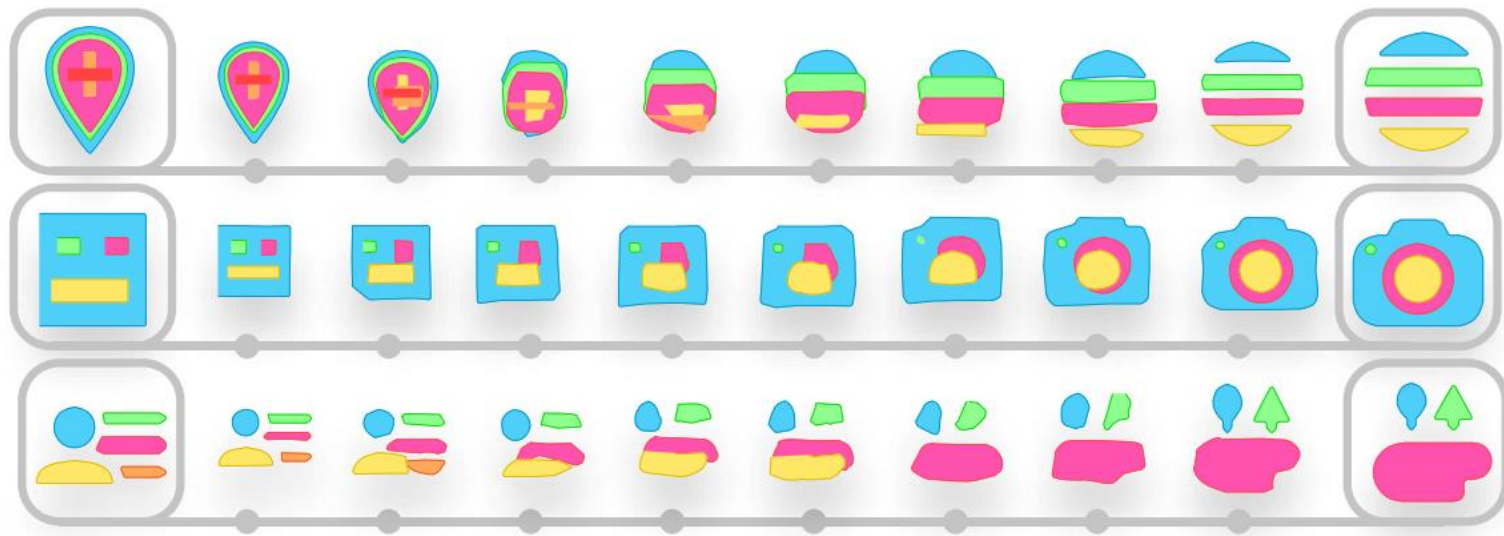
- Эмбеддинг команды:
 - Эмбеддинг для типа команды (<SOS>, M, L, C, Z, <EOS>)
 - Эмбеддинг для координат (вектор из 6 чисел)
 - Позиционное кодирование команд

DeepSVG (2020) – общая архитектура



DeepSVG (2020) - результаты

0 1 2 3 4 5 6 7 8 9
a b c d e f g h i j k l m n o p q r s t u v w x y z
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z



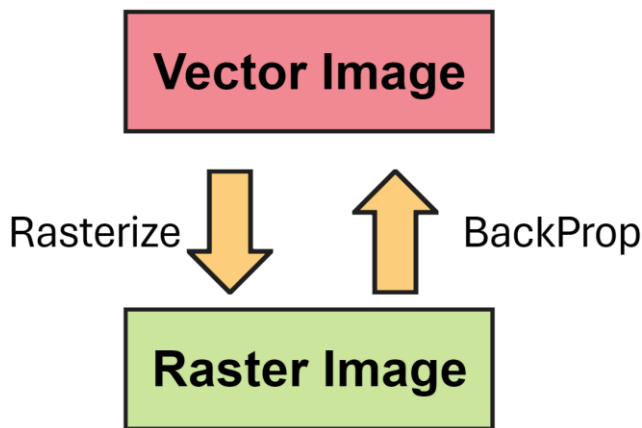
Текущие проблемы

Внутреннее ограничение на кол-во путей и команд.

Нужен датасет из векторных изображений.

Собрать его – дело непростое.

DiffVG (2020)

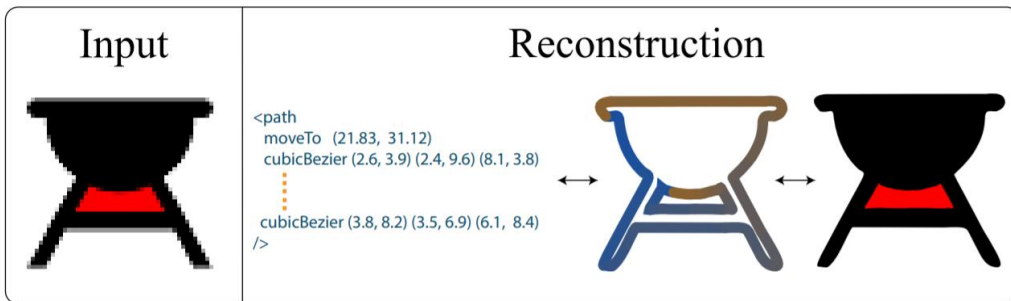
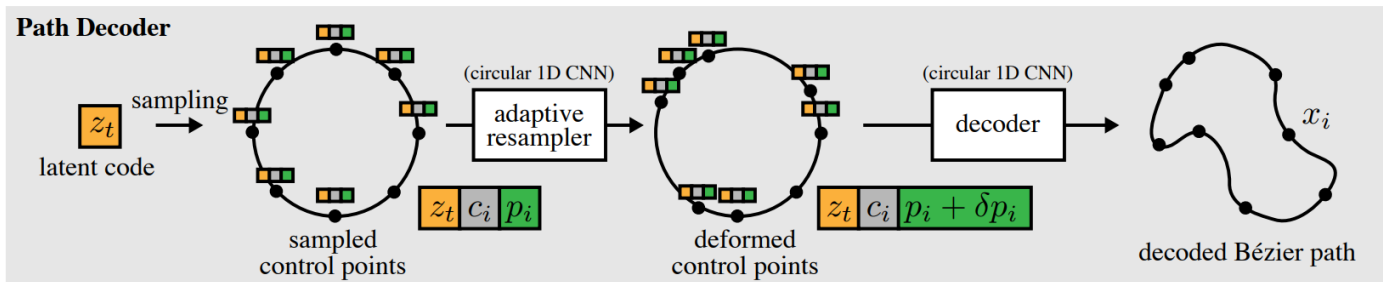
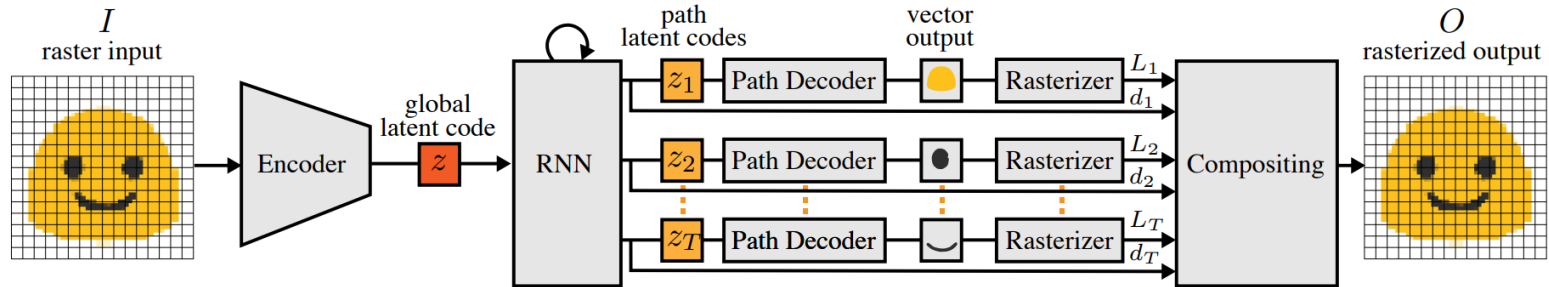


Дифференцируемый растеризатор векторной графики

Позволяет «связать» векторную и растровую графику через обратное распространение ошибки.

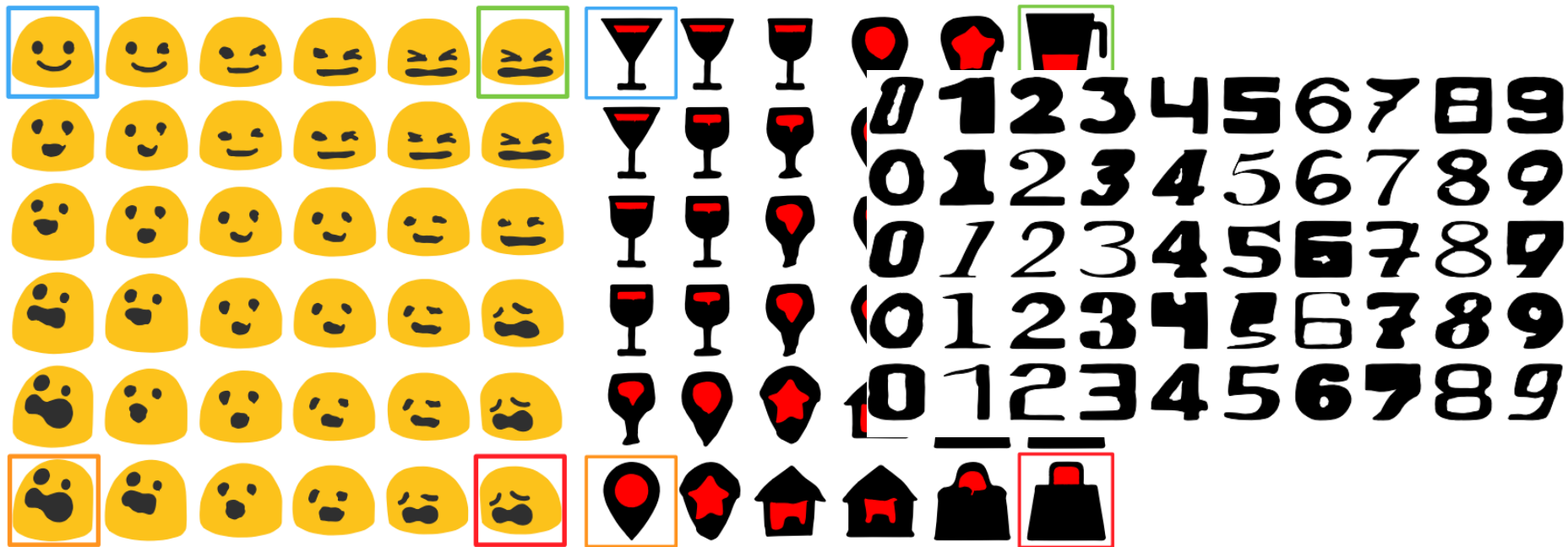
Можно сконструировать векторное изображение, растеризовать его и эту растеризацию использовать в функции потери.

Im2Vec (2021) – VAE векторизатор растровой графики

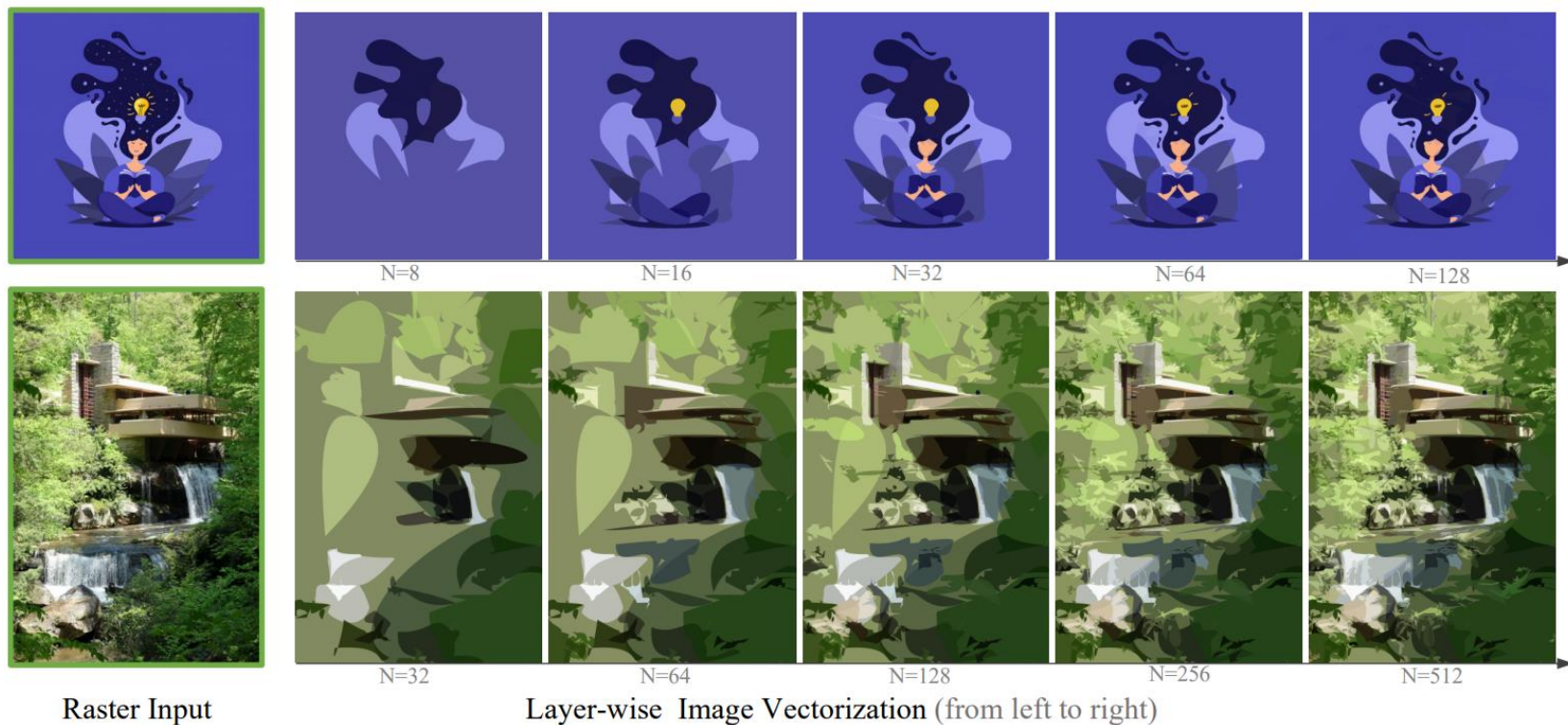


<https://arxiv.org/pdf/2102.02798.pdf>

Im2Vec (2021) – результаты



LIVE (2022) – векторизатор растровой графики



<https://arxiv.org/pdf/2206.04655.pdf>

Достоинства:

- Введен лосс на самопересечение фигур (*)
- Можно получать векторизации картинок с заданным количеством фигур

Недостатки:

- Работает чрезвычайно долго

CLIPDraw (2021)

Генерация SVG
изображения по
входному тексту

“A drawing of a
cat”.



“A painting of a
sunset”.



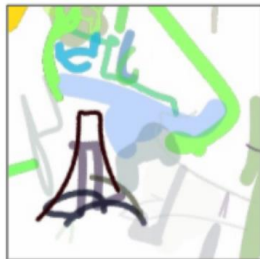
“Underwater”.



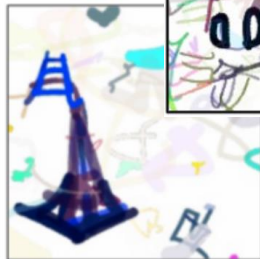
“Sheep wearing
a top hat”.



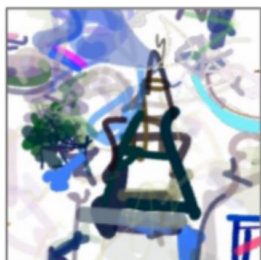
16 Strokes



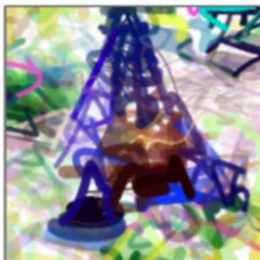
32 Strokes



64 Strokes



128 Strokes



256 Strokes



512 Strokes

Параметры:

- число кривых
- кол-во итераций
- жирность обводки (нет fill)

<https://arxiv.org/pdf/2106.14843.pdf>

CLIPDraw (2021) - архитектура

