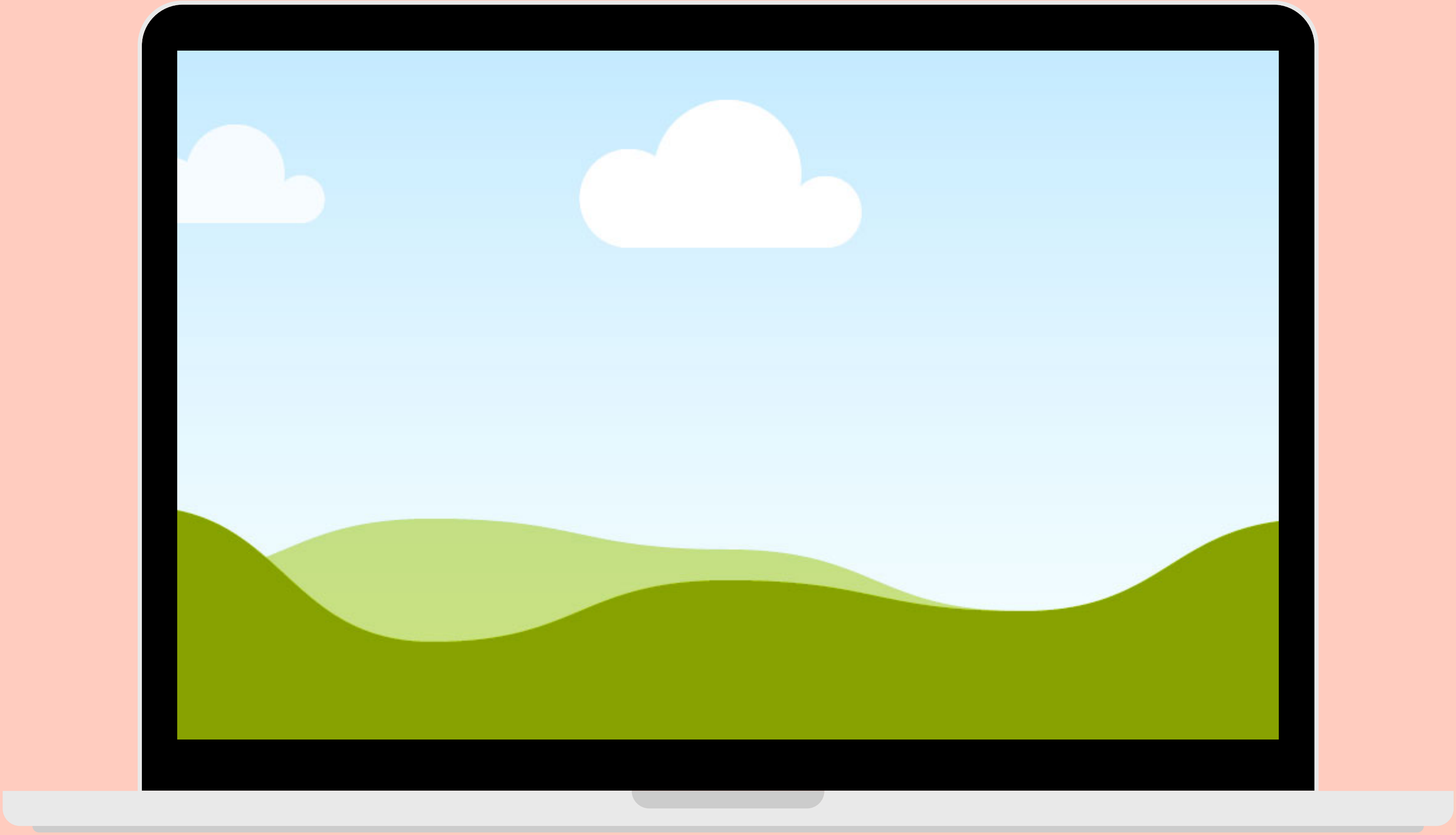
Init2

Anonymization



Hello everyone.

Abhishek_Rajora
brillard1

Abu_Shahid
ceyxasm

Problem statement-I

- **Surveillance cameras:** Surveillance cameras are becoming ubiquitous in public spaces, leading to concerns about privacy violations and misuse of data.

- How can we protect individuals' identities while still using the footage for security and analysis purposes?

Facial Features

- Facial landmarks
- Facial symmetry
- Facial proportions
- Texture analysis
- Facial expressions
- Age estimation
- Gender recognition

SOLUTION

- We propose an AI-based face anonymizer that can detect and blur faces in surveillance footage, while preserving the statistical information about them.

- Explored the deep maths behind the blurring techniques and explain why they are suitable for face anonymization.

Dataset used

1. Utilized large and diverse data set of face images collected from various sources, such as public domain databases, online platforms and real-world surveillance footage
2. Covers different scenarios, such as varying lighting conditions, different angles and poses, multiple faces in a frame, occlusions and accessories
3. Face Detection Data Set and Benchmark (FDDB) and Labeled Faces in the Wild (LFW) were the primary dataset on which the hyperparameters were finetuned on
4. These weights are used in mediapipe library for our purpose

Design Applications

- ***Goal***: Perform face detection and apply one of the three blurring techniques: Gaussian Blur, Pixelated Blur or Bilateral Blur to blur the faces while still preserving the statistical features.

- ***Application***: The technique is applicable and useful to any organization or entity that uses surveillance cameras, such as government agencies, law enforcement, transportation, retail, healthcare, education, etc.

- The algorithm can balance the need for surveillance with the respect for privacy, and comply with the relevant laws and regulations.

Algorithm Design

- ***Face Detection***: A single-shot detector (SSD) architecture with a MobileNetV2 feature extractor.

$$f_d(I) = \{B_i, L_i\}_{i=1}^{n}$$

- ***Blurring Techniques***
  - ***Gaussian Blur:***
    - *smooths out the pixels by averaging them with their neighbors using a Gaussian filter, the degree of which depends on the size of the filter*
    - *simple and fast, but it may lose some details and edges of the face*

$$f_g(I, B_i) = I * G_\sigma$$

Algorithm Design

- ***Pixelated Blur:***
  - *divides the face region into small blocks of pixels, and replaces each block with its average color, degree of which depends on the size of the blocks*
  - *Simple and fast, may create artifacts and distortions in the face region*

$$f_p(I, B_i) = \frac{1}{N^2} \sum_{j=1}^{N^2} I_j$$

Algorithm Design

- ***Bilateral Blur:***
  - *smooths out the pixels by averaging them with their neighbors using a bilateral filter, but also preserves the edges by weighting them according to their similarity*
  - *The degree of blurring depends on two parameters: spatial distance and color difference*
  - *more complex and slow, but it preserves more details and edges of the face*

$$f_b(I, B_i) = \frac{\sum_{k \in N_i} w_s(||k - i||) w_c(||I_k - I_i||) I_k}{\sum_{k \in N_i} w_s(||k - i||) w_c(||I_k - I_i||)}$$

Algorithm Design

- ***Bilateral Blur:***
  - *Based on the architecture of Variational Auto Encoders (VAE) to reduce/encode faces into preserved feature dimensions*
  - *And finally reconstruct/decode Faces based on preserved statistical features*

Keypoints and Remakrs

- *Uses state-of-the-art face detection and blurring techniques to achieve high performance and quality*
- *Preserves the statistical information about the faces even after blurring them, which can be useful for security and analysis purposes.*
- *Integration of our algorithm (backboned at either local machine or cloud server) with a web or a mobile application can allow users to upload or stream surveillance footage to extract anonymized features*
- *The application can also display the statistical information about the faces in the footage using charts or graphs*
- *Allows users to choose between different blurring techniques and adjust their parameters, and see the results in real time.*
- *And finally it balances the need for surveillance with the respect for privacy, and complies with the relevant laws and regulations*

Problem statement-II

- Financial reports are generated in huge numbers from various departments.
- These data are needed by corporations and companies for analysis and inference.
- They can expose the privacy of an individual if not appropriately handled.
  - To this end, we present 4 small yet impactful implementations to better handle financial data.

Dataset used

1. Housing data.csv: Listing of 545 housing property with crucial details such as price, number of rooms, area, etc.
2. VL2G_employees data: data of 57 employees with their names, IDs, email address etc.
3. Company_tax data: data of 109 companies with their age, founder, profitability, and tax filing status
4. Offices data: number of offices of various companies.
5. Adults data: data of 45k adults

v0- Numerical data manipulation

- anonymize
- binarize
- clamp
- categorize
- fill
- min-max scale
- remove
- round

SALIENT FEATURES

- Gives complete control to the end user as to which features to anonymize and which features to preserve.
- While anonymization, we may chose whether we want to retain statistical properties of the features of not.

v1.0- Named data manipulation

- name last_name first_name
- email phone_number
- zip-code street street_address city
- iban-number
- text sentence

SALIENT FEATURES

- Gives complete control to the end user as to which features to anonymize and which features to preserve.
- Cannot be reversed.
- Same ***input labels*** become same ***output labels***

v1.2- Named data manipulation across multiple data files

- name last_name first_name
- email phone_number
- zip-code street street_address city
- iban-number
- text sentence

SALIENT FEATURES

- Gives complete control to the end user as to which features to anonymize and which features to preserve.
- Cannot be reversed.
- Same ***input labels*** become same ***output labels*** and extended across multiple files
- Data integrity is thus preserved.

v2- Implementing and visualizing k-diversity

- Sensitive attribute: Refers to private or confidential information in a dataset.
- ***Goal***: Achieve k-diversity by ensuring each group/partition in the anonymized dataset has at least k distinct values for the sensitive attribute.
- ***Anonymization*** techniques: Generalization or suppression to protect the sensitive attribute.
- Examples of sensitive attributes: Medical conditions, race, religion, financial information, etc.
- ***Purpose***: Reduce the risk of re-identification or inference of sensitive information about individuals.
  - optional hashing for even better security of sensitive columns

This was init2 :) Thank You