Image Quilting & Texture Transfer

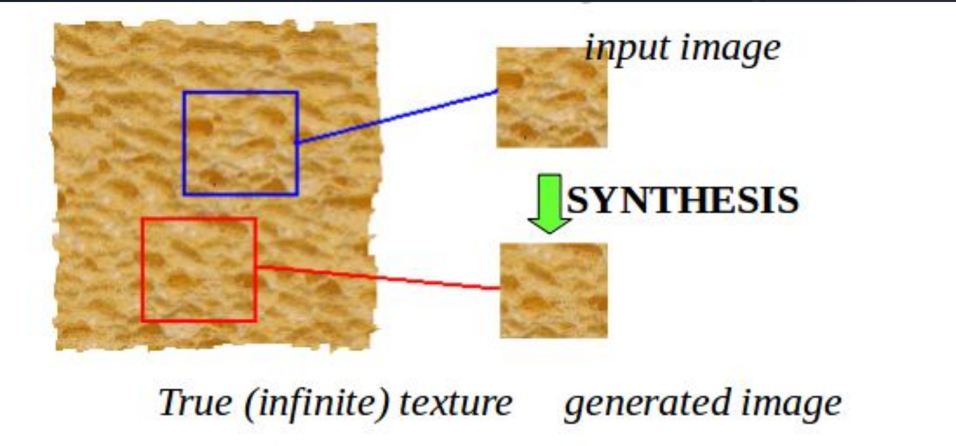
Team **N**ame	**T**he**A**rt**A**dda
Project **ID**	**44**

***M**embers*

Anvesh **C**haturvedi *2016109**4***

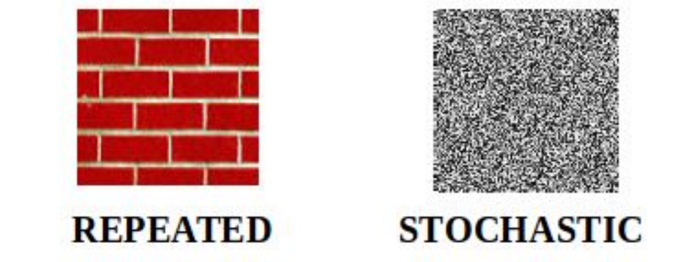
Yudhik **A**grawal *2016109**3***

Goal *Of* **Texture Synthesis**



Given a *finite sample* of *some texture*, the *goal* is to ***synthesize other samples from that same texture***.

Challenges



- *Modelling* the whole spectrum ***from repeated to stochastic textures***.
- **Texture Analysis** : *Getting an idea of the texture structure* for *correctness of texture synthesis.*
- *Finding the nature* of the image and *adjusting the parameters* of the algorithms accordingly.
- Multiple varieties of textures [*Without any pattern*].

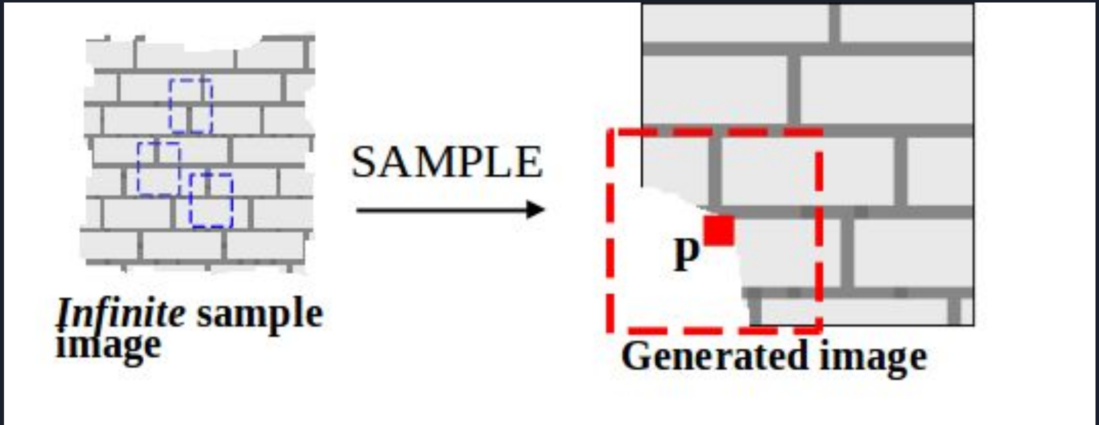
Texture Synthesis *by* Non-parametric Sampling

- *Efros & Leung, 1999*

Approach

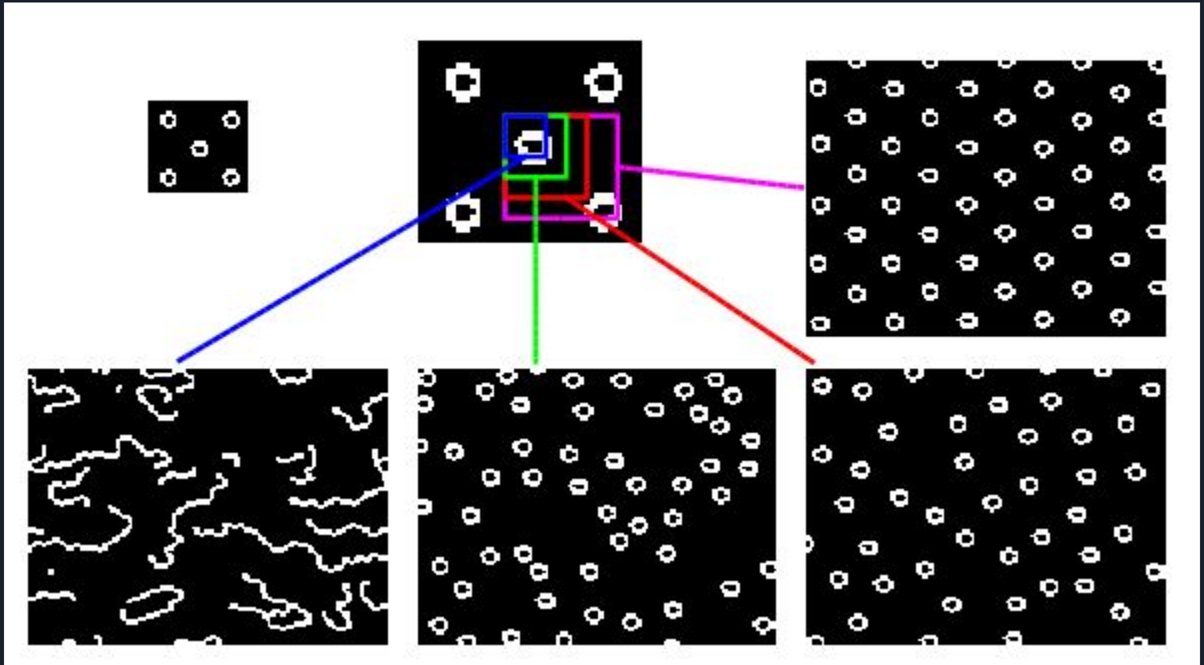
- The *main goal* of the ***texture synthesis algorithm*** is to **preserve local structure** at all stages of the algorithm.
- We need to fulfill certain constraints at each stage to ensure robustness of the results.
- ***One pixel at a time***. Our algorithm "*grows*" texture, *pixel by pixel*, *outward from an initial seed.*
- The *neighborhood of a pixel* is ***modelled*** as a *square window* around that pixel. The *size of the window* is a *free parameter* that ***specifies how stochastic*** the user believes this texture to be.

Synthesizing *One* Pixel



- We assume the **Markov property** to be true *where we can find* the ***conditional probability distribution*** of any point *'P' given* that *we have its neighbourhood.*
- We *find* the *best match* using **SSD error** (*weighted by a* ***Gaussian*** *to emphasize local structure*), and *consider all samples* in *some range* of *distance from that match.*

Randomness Parameter – *The Window Size*



The ***size*** *of the neighbourhood window is a parameter that specifies how* ***stochastic the user believes this texture*** to be. *Different outputs are observed as we go on changing the* ***window size***.

Image: Fence

Window Size - 21

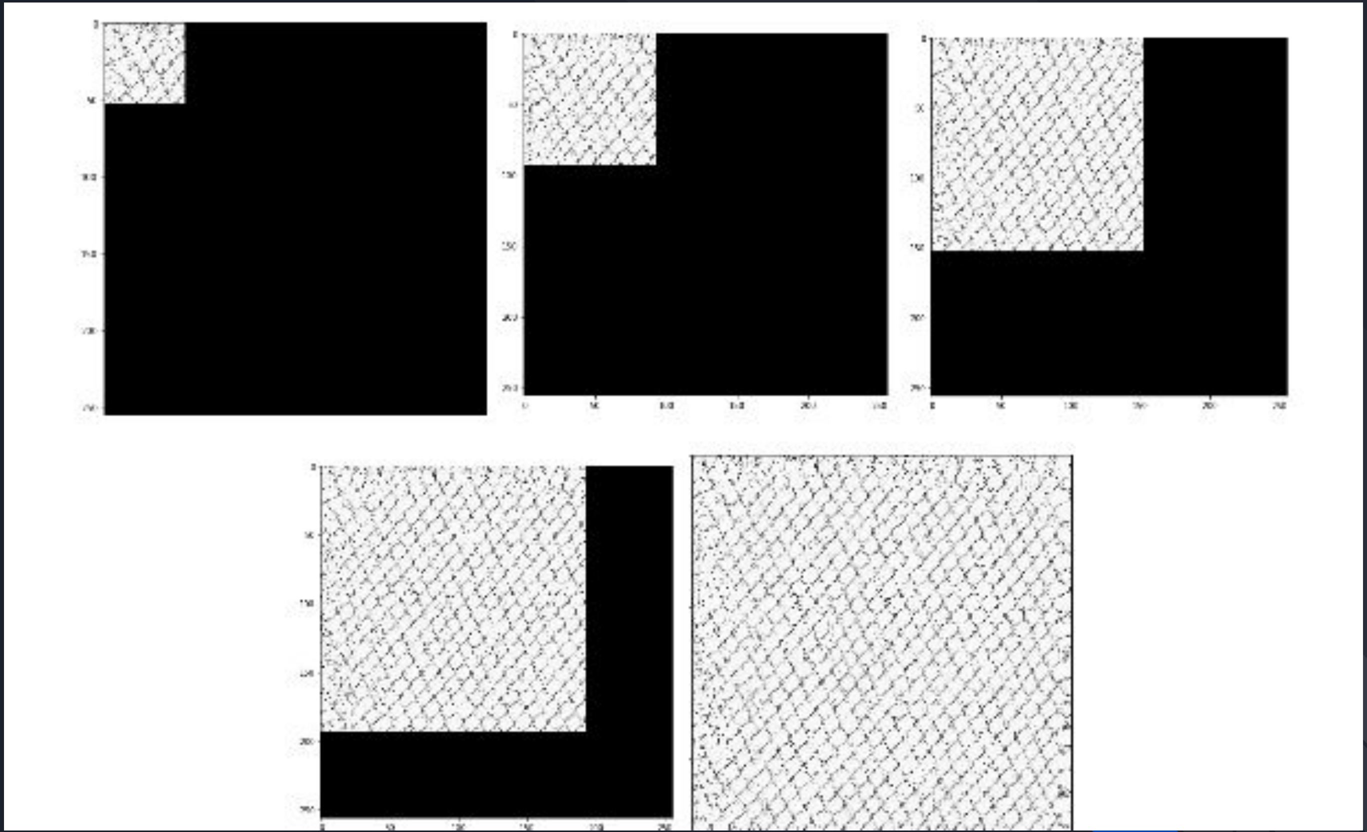


Image: Fence

Window

Size - 23

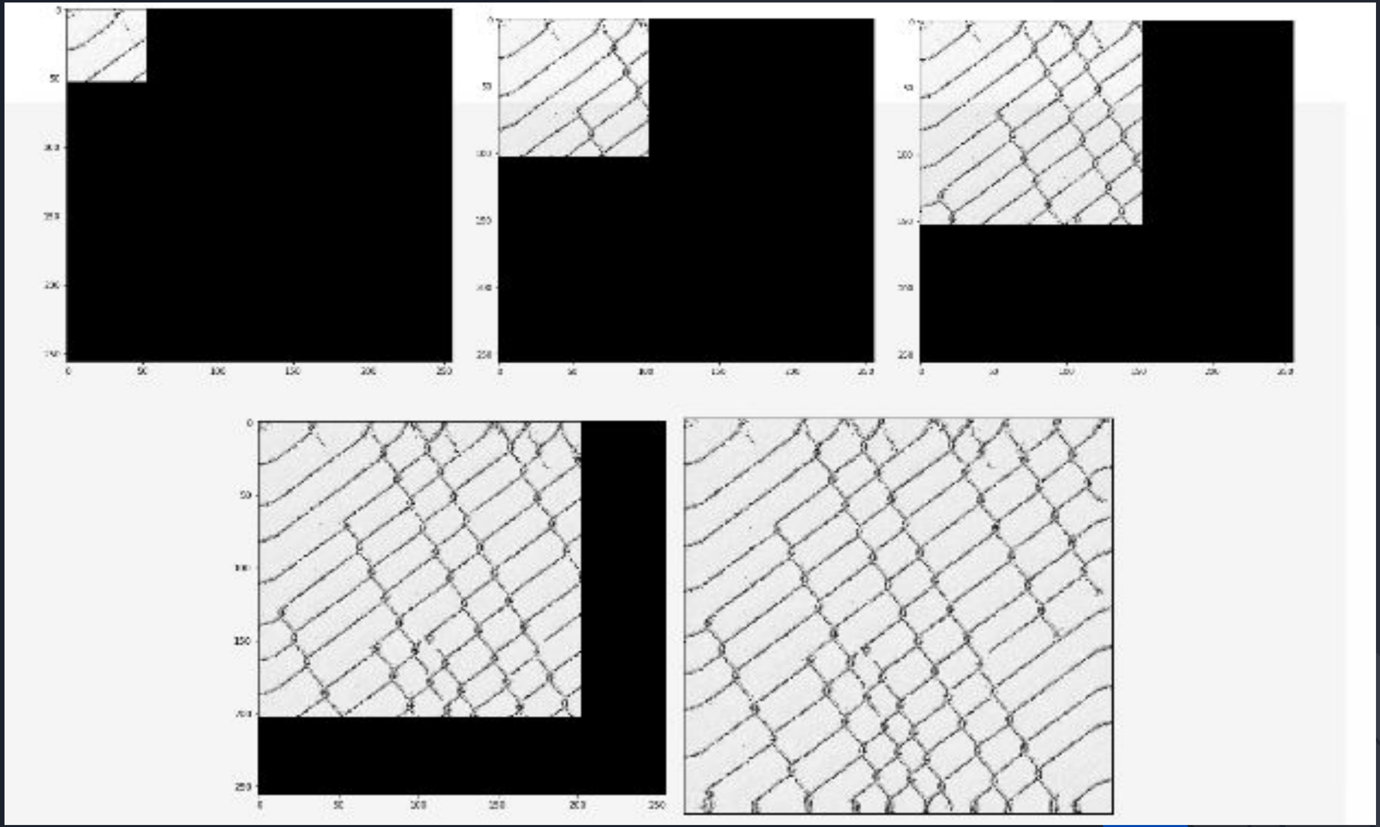
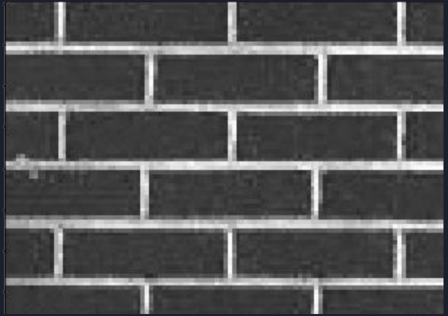


Image: Brick

Window Size - 21



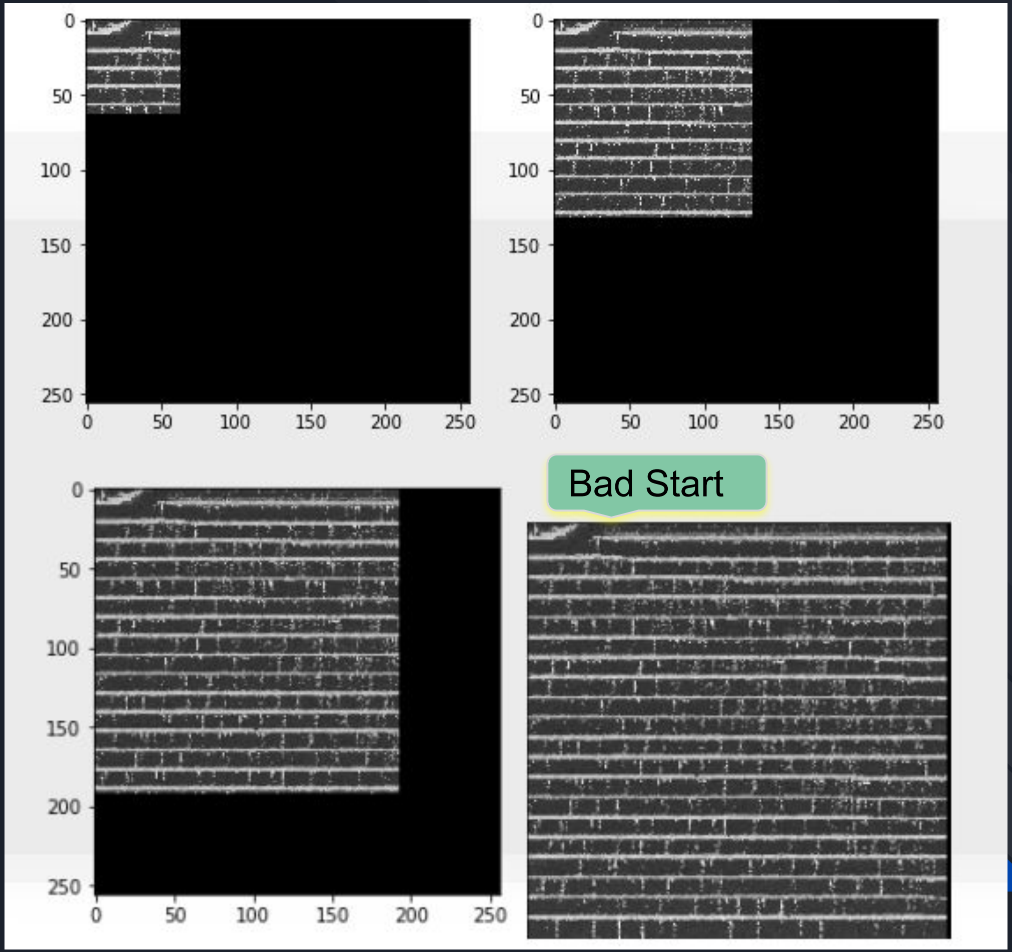
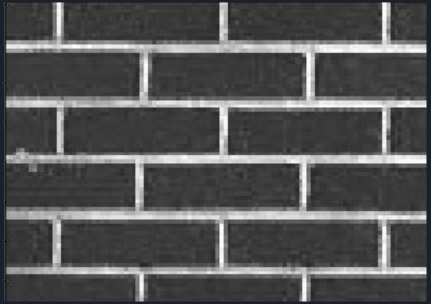
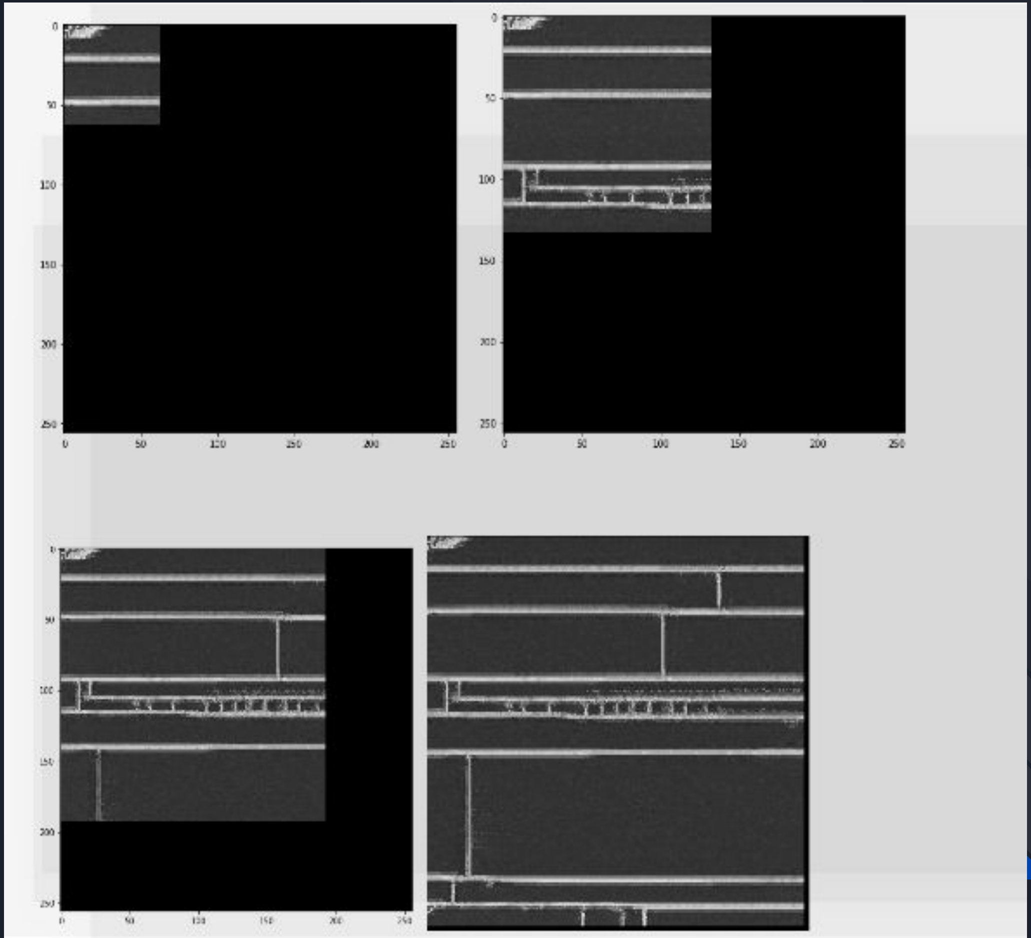


Image: Brick

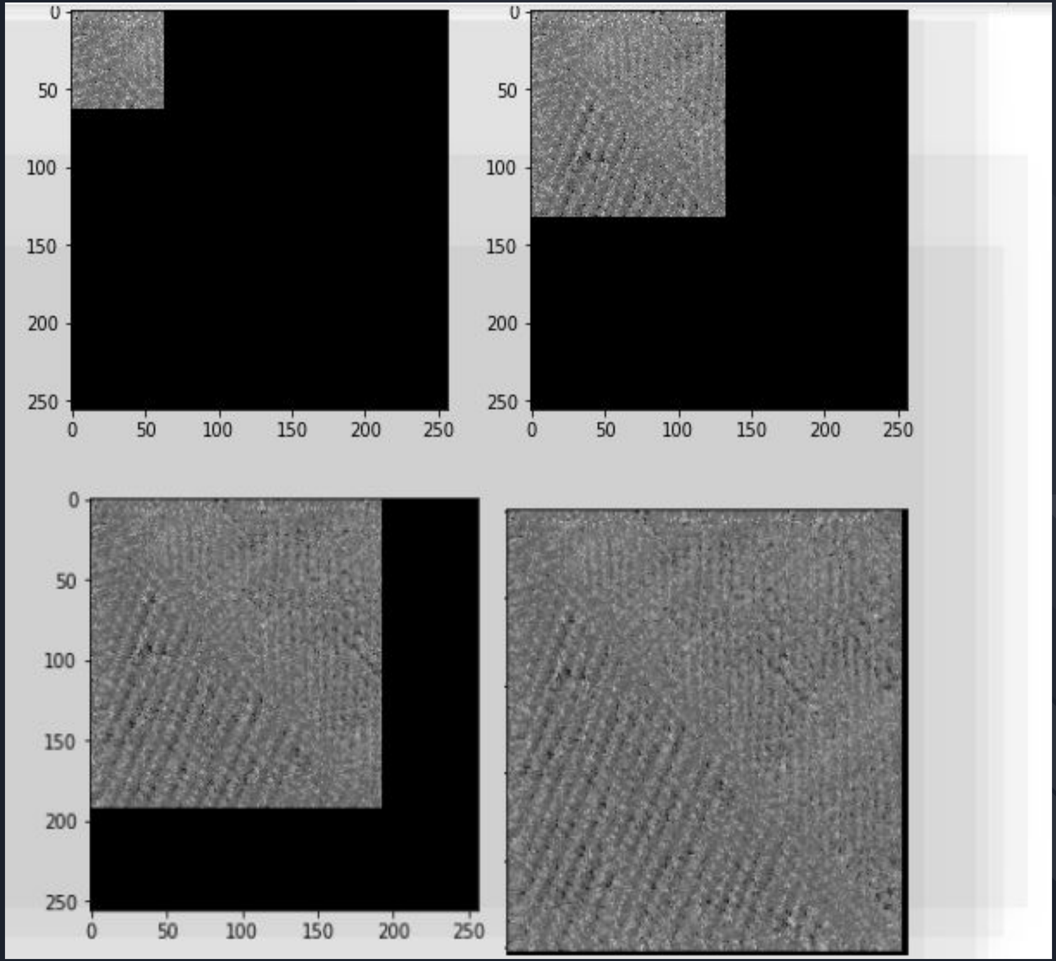
Window

Size - *15*





Not So Good Case



Challenges

- *One* **problem** *of the algorithm is its tendency for some textures to occasionally* '***slip***' into a *wrong part of the search space* and ***start growing garbage*** or *get locked* onto one place in the sample image and *produce verbatim copies of the original.*
- One of the *biggest challenges* of this approach is the *hyperparameter window-size* which *greatly influences* the ***output characteristics and quality***.
- ***Greedy***, ***slow*** and ***heuristic*** based *approach.*

Advantages

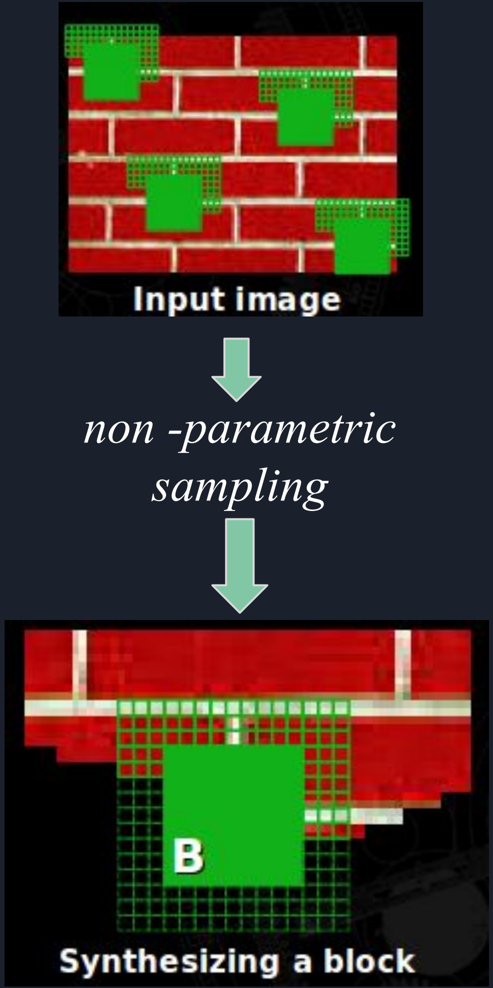
- *Conceptually simple approach, easy to* ***understand*** and ***implement***.
- Has the *capacity* to ***model a wide-range of real world textures***.

Image Quilting *for* Texture Synthesis & Transfer

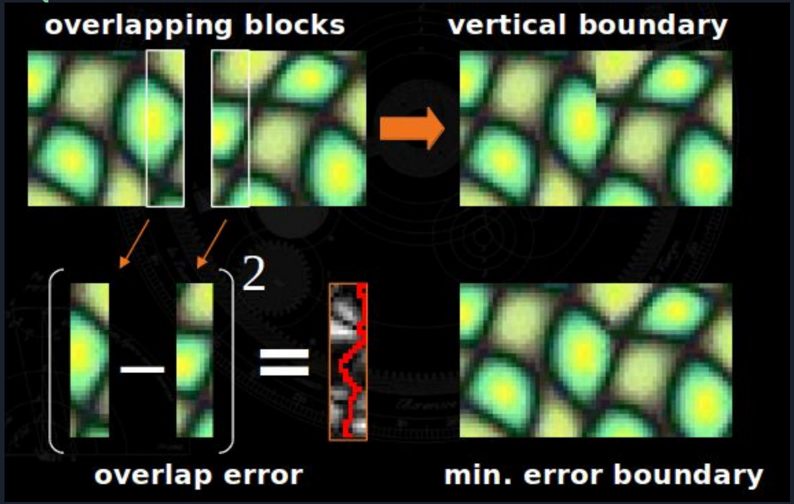
- *Efros & Freeman, 2001*

Idea

- For ***most complex textures*** very *few pixels* actually have a *choice of values* that can be *assigned to them*. That is, during the *synthesis process most pixels* have their values **totally determined** by what has been synthesized so far.
- Taking this into *consideration*, we ***define the unit of synthesis*** in this approach to be a *block* **Bi**.



Minimal Error Boundary Cut



B_1 *and* B_2 are *two blocks* that *overlap* along their ***vertical edge*** with the regions of ***overlap*** B_{ov1} *and* B_{ov2} , respectively, then the *error surface* '**e**' and *cumulative minimum error* '**E**' for all paths is calculated as:

$$e = (B_{ov1} - B_{ov2})^2$$

$$E_{i,j} = e_{i,j} + \min(E_{i-1,j-1}, E_{i-1,j}, E_{i-1,j+1})$$

$$MinPaths_{i,j} = \text{argmin}(E_{i-1,j-1}, E_{i-1,j}, E_{i-1,j+1})$$

Results

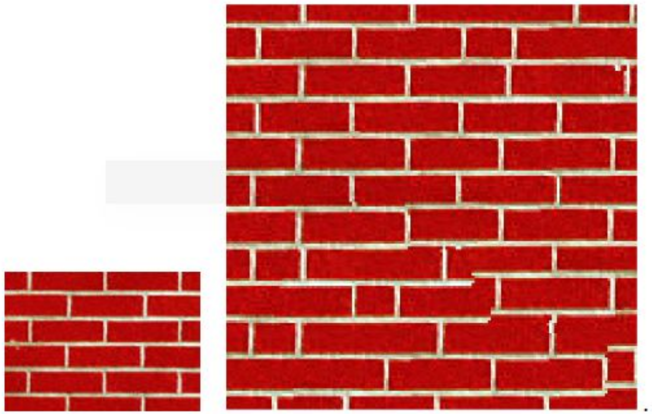
Factor = 2

Error Threshold = 0.15

Factor = 2

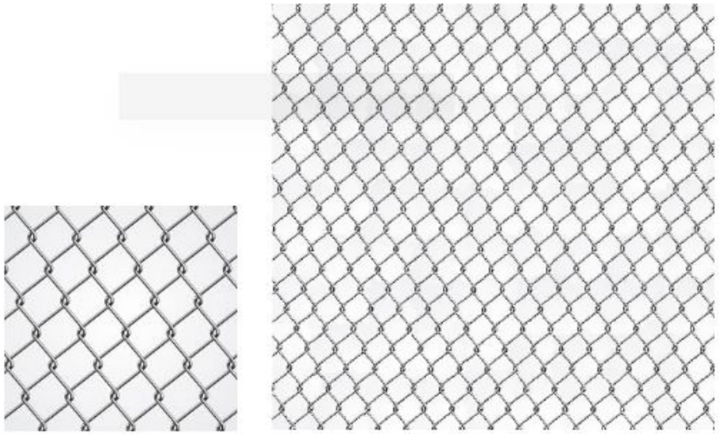
Error Threshold = 0.05

Results



Factor = 2

Error Threshold = 0.05



Factor = 3

Error Threshold = 0.15

Texture Transfer

- *Efros & Freeman, 2001*

<u>Basic Idea</u> : Take the **texture** *from* **one object** and ‘***paint***’ it *onto* ***another object.***

This requires separating texture and shape.

Approach

- The ***synthesis algorithm*** has been *augmented* by *adding the requirement* that *each patch* must *satisfy a desired correspondence map as well as satisfy the texture synthesis requirements.*

Challenges

- *Image being* ***synthesized*** must *respect two independent constraints:*
    - The *output* are *legitimate, synthesized* examples of *the source texture.*
    - The *correspondence image* mapping must be *respected* for ***proper texture transfer*** observations.
- Because of added constraint, *generally one synthesis pass* through the *image* is **not** *enough* to ***produce a visually pleasing result***.
- *A* ***weighted error*** giving appropriate weights to the *two constraints* needs to be defined. The *way in which value of* α*(the weight parameter)* is decided *can alter the results* to some extent.

Overlay

Iteration - 1

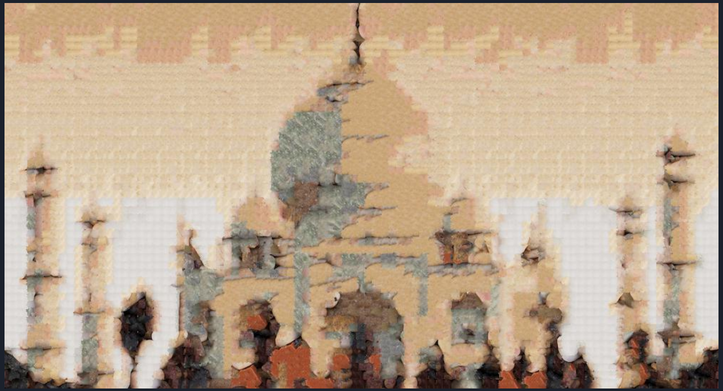
Iteration - 2

Iteration - 3

Iteration - 4

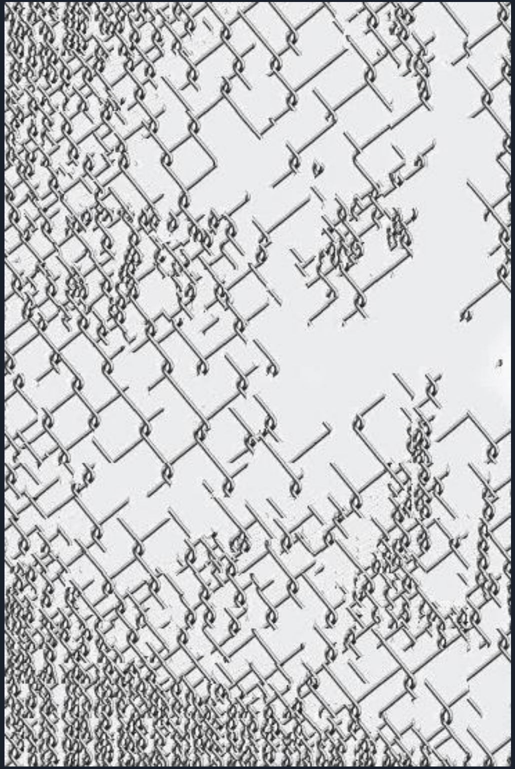
Iteration - 5

Iteration - 6



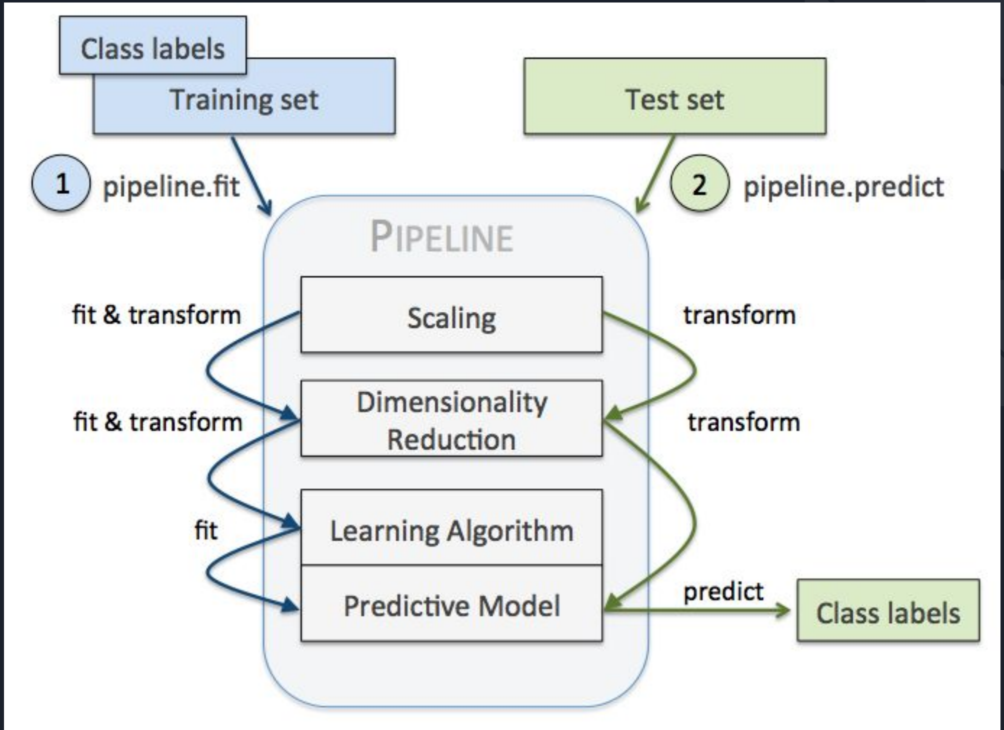
Iteration - 7
Final Output

Another Example



Further Experiments/Uses

Pipeline to exploit how efficiently our algorithm can help in increasing dataset.



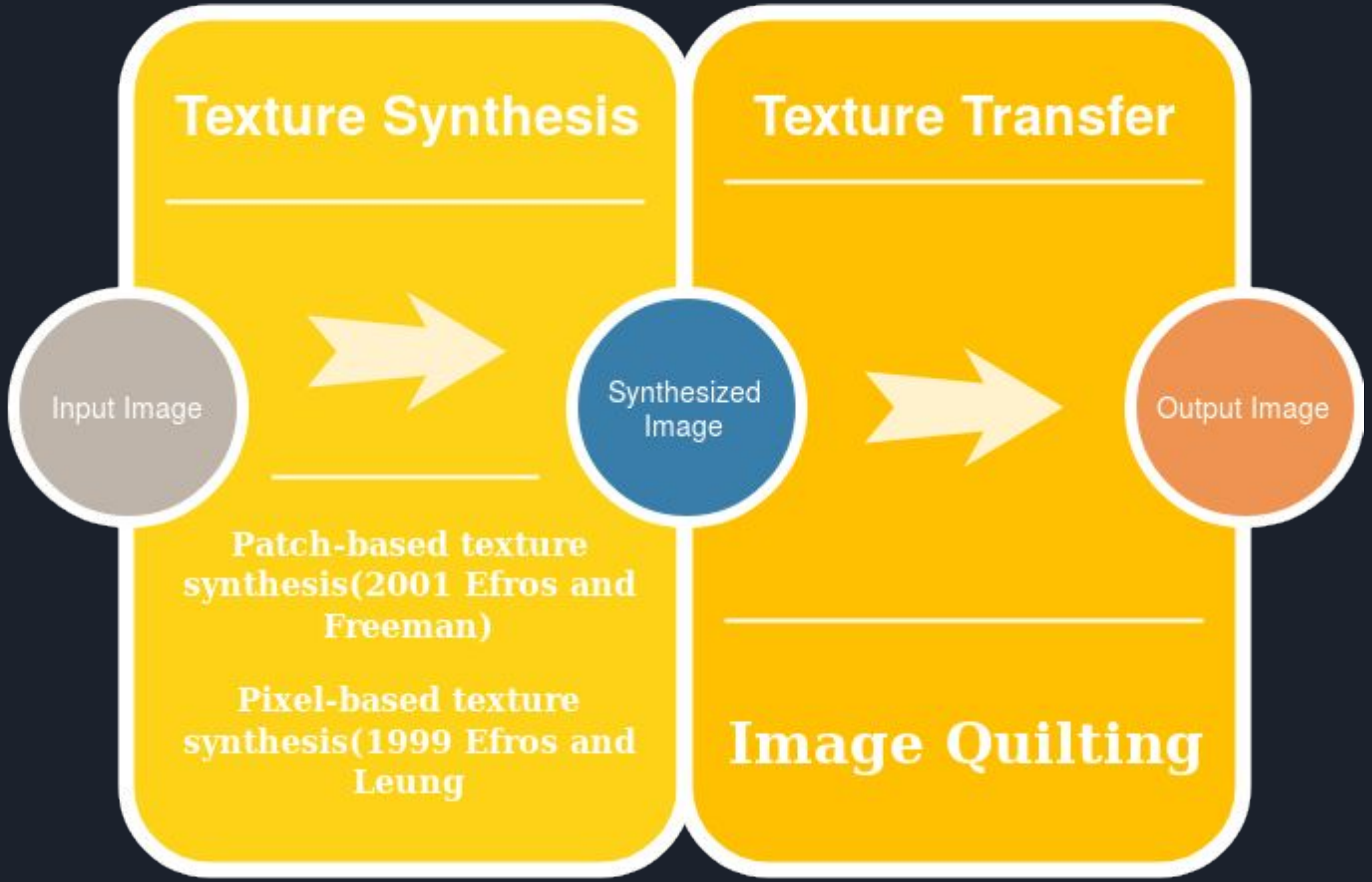
Extending Database!!! for Various texture images

Our ***Texture Synthesis*** provides images with ***increased pattern size*** and the *Artifacts* produced during the process are very small which can be ignored.

Idea : *Various Texture images* were *synthesized* from our image synthesis algorithm and a classifier is created for various texture images[in our case 3] using the ***Linear SVM classifier***.

- *Multiple random images* are generated from the *given* texture images and are *split* into ***training data*** *and* ***testing data***.
- PCA is done for ***pre-processing*** before *training the classifier*.
- We *observe an accuracy* of ***100%*** on the test set showing the ***robustness*** of the algorithm *via* ***the great separability*** of the *data generated* using the outputs.

We get ***100% accuracy*** not just while training the data without reduced complexity ***but even after taking the PCA with number of components as 8***.


Texture Synthesis
Texture Transfer
Input Image
Synthesized Image
Output Image
Patch-based texture synthesis(2001 Efros and Freeman)
Pixel-based texture synthesis(1999 Efros and Leung
Image Quilting

Hasta la Vista