Deepstamp - whitepaper and roadmap

A stuttering man trying to win success and a bundle of money and yells for more.

What is deepstamp?

Deepstamp is a blockchain of AI generated art. You can see an example of a deepstamp on the cover page of this document.

Deepstamp has following properties:

- **Mathematically scarce art:** only one artwork gets generated per block and that is why nobody can increase the supply of artwork in the network. (This is unlike NFTs which anyone can create in any number of quantities).
- **Decentralized:** nobody controls the system. Anyone can participate in creating AI-generated art by running the protocol (or software that implements the protocol) on their computer.
- **Fully-autonomous art:** since an AI generates art, there's no human creative input required. Nevertheless, the art produced by the deepstamp network is often beautiful and evocative.
- **Ownership of art:** just like cryptocurrencies like bitcoin, you can own, transfer and exchange the generated art (fully or partially) with anyone else.
- **Environment friendly, "Proof of Art" (PoA) algorithm**: the PoA algorithm allows for generation of art in trustless, decentralization manner by borrowing from environment friendly "Proof of Stake" algorithms and traditional "Proof of Work" algorithms.

One way to interpret deepstamp is that it's a cryptocurrency where each coin has a unique look. This aspect helps the owner of a deepstamp in two ways:

- **Exhibit his/her collection:** since bitcoin is fungible (all of them look the same), an owner of bitcoin has nothing to show for except the number of bitcoins s/he owns. In contrast, a deepstamp owner can show not just number of deepstamps owned but also what types of deepstamps s/he owns.
- **Charge a premium:** If the owned deepstamp is particularly attractive or appealing, the owner of it can easily demand a premium for it. Contrast this with bitcoin, where all bitcoins have the same value.

From the description above, it can be easily seen that deepstamp is not a direct competitor of either bitcoin or NFTs. It's different from both of them and hence we believe deepstamp represents a totally new and unique kind of crypto-asset.

Overview of the deepstamp network

At a very high level, deepstamp works in the following manner:

- Like most cryptocurrencies, the backbone of deepstamp is a blockchain
- This blockchain is replicated by participating nodes in the network
    - Since deepstamp is decentralized, anyone can participate in the network
- Blockchain is a sequence of blocks, which is a data structure with records of ownership of generated images
    - The ownership is linked to a pair of cryptographically generated public/private keys

- Valid blocks get generated when a node (that's amongst a set randomly chosen from the network) generates a "good enough" image for a (sentence) prompt that's derived from hash of previous block data (see next section for details)
- The transactions between nodes are transmitted to the entire network and the nodes that found a valid block can include them in the block (after verifying that the transactions are valid, in the sense of nobody is transferring deepstamps they don't own). Such a block is called a *painted* block.
- The painted block is transmitted to the entire network to be included in the blockchain
    - All nodes verify the painted block again that it meets the criteria of "good enough" image and the included transactions are valid
- The entire process starts again with generation of the next painted block
- In case due to latency, several inconsistent chains emerge in the network, the longest chain among them is considered to be the right one (proof-of-work style resolution of consistency)
    - This is why a transaction should be considered valid if there have been at least 6 additional painted blocks in the chain after the transaction painted block

Details of various components of the blockchain

How is the autonomous prompt for the art generated?

- The prompt for the current block is a sentence depends on the hash of the contents of the previous block so that nobody can generate such sentence in advance (without knowing hash of the latest minted block)
- Proposed method to generate sentence:
    - Take top 10 English language books from Gutenberg.org
    - Input the the raw text from these books into a markov model
        - In v0.01-alphanet, Markovify library is used
    - Set the random seed to hash of the previous block's data
        - E.g. random.seed(hash)
    - With this random seed, generate the first sentence from the markov model that has minimum 40 and maximum 80 characters
        - Try this for 100 times
    - If this succeeds (which most likely will), you have a sentence for the block. Examples of sentences generated through this method:
        - A stuttering man trying to win success and a bundle of money and yells for more.
        - This Cook Book was invented by a scratching noise on the outside.
        - Cupid is ever somewhat to remember; others lay criticism asleep by a child.
        - A mythical place in a strange house.
    - These sentences almost make sense, and sometimes they exactly make sense

        - If in 100 tries, markov model isn't able to generate a sentence that satisfies min and max criteria
            - Skip the block, adding previous block's hash as a data (so as to provide a unique hash for the next block)

How is art created for the generated prompt?

The key task in the system is to generate a 512x512 colored image (in PNG format) that matches the prompt.

- We use two deep learning models (Google's BigGAN combined with OpenAI's CLIP) to guide generation of an image, given the block sentence
    - There is a code called BigSleep that does exactly this. We are going to reuse that code in v0.01-alphanet

How is art created validated to be "good enough"?

We use an algorithm called "Proof of Art". It goes as follows.

For a given 512x512 colored image and a text prompt, we calculate image quality as follows

- We use CLIP, the deep learning model by OpenAI for this task
- Given a prompt and an image, CLIP provides a probability (expressed in terms of logits) that image and text match
    - Logits range from -infinity to +infinity, but in practice for any given image and text, CLIP outputs a logit score between +15 to +50.
    - Manual analysis suggests that a score of 40+ usually a good quality image that represents a given text

A simple way to determine whether an image is "good enough" is to simply check if the image's logit is >40 for the given text. If this approach is adopted then the node which is first to generate an image with a logit of >40 gets to paint (mine) the block.

However, there are a few challenges with this:

- Even though it's unlikely, but it seems possible that for a given prompt, no image exists for which CLIP model outputs a logit > 40
    - In such a case, the blockchain will halt as no block could be finalized
- It could take a long time for (selected) nodes to generate an image that qualifies the "good enough" criteria of >40 logit.

Since generating a "good enough" image could take an unbounded time, if we want the blockchain to keep finalizing blocks at a deterministic rate, we have to keep decreasing the threshold of "good enough" with time, so that as time -> infinity, the probability of finalizing an image as "good enough" -> 1.

WRITE ABOUT OCR

This decrease of threshold with time is solved as follows:

- The initial threshold for "good enough" is 40

- For an image that's NOT crossed this threshold to considered valid by the network, it has to be accompanied with a proof of work with a difficulty inversely proportional to how far away is the image from the threshold
    - This proof of work happens in a background thread and is exactly same as the bitcoin network
    - As a quick overview, this means that a number called nonce has to be found by the node which when hashed together with other metadata of the block (such as block height) produces a hash less than a particular number as determined by the difficulty (in bits)
- So, to summarize an image can be considered "good enough" with a lower threshold of logit if it's accompanied by a proof of work of a higher difficulty than the targeted difficulty
    - For each logit reduction from 40, the difficulty for proof of work increases by 1 bit
- The difficulty auto-adjusts every 10 blocks (only including the blocks where proof of work was used)
    - Difficulty targets an average rate of 1 PoW block / 10 minutes (like bitcoin)

Image logit	**Difficulty level (in bits)**
>=40	0
39-40	Current difficulty
38-39	Current difficulty + 1

How are eligible nodes for painting the block selected?

Nodes that are interested in painting (mining) blocks lock their coins in a special type of transaction that makes such coins unspendable. For each block, the set of nodes with locked coins are randomly shuffled. The random seed for this shuffling comes from the previous block's hash (so that it's different for each block). After shuffling, a maximum of 100 or 1% of all nodes are chosen with a weight equal to the number of coins they've staked (so that higher stake, the higher the chances of painting a block). These nodes are eligible for making the next block.

If there are not 1000 nodes with locked coins, the network reverts to all nodes eligible for creating a coin.

What does the coin supply look like?

Like bitcoin, a total of 21 million deepstamps will be created and each deepstamp will have. Each deepstamp consists of 100 million deeps (equivalent of satoshi). The halving period of supply is exactly like bitcoin.

This means an owner of deepstamp has two properties:

- Coin number (# of deeps)
- Coin quality (the kind of deepstamp this coin was originally painted with)

Hence, the value of deepstamp will depend both on quantity and quality.

Run a node in the prototype chain

A very early prototype of deepstamp is running with a few nodes. To join the network, fill the form and we will send instructions (you need gpu)

You can see here all the deepstamps that the prototype chain is creating.

Join the launch community

If you want to join the initial community that will launch deepstamp, fill the form here. If you have the right skills to contribute, we will email you a discord channel link.

Skills we're looking for:

- Blockchain developers
    - To select the right framework to launch the blockchain
        - Should we write from scratch or use an existing framework like Matic, Cosmos or Polkadot?
- Protocol designers
    - To think of flaws in proposed network and think of improvements
- Visual designers
    - For developing website and other artwork
- Infrastructure
    - Those who can run the node for prototype chain, needs GPU

Benefits of joining the launch community

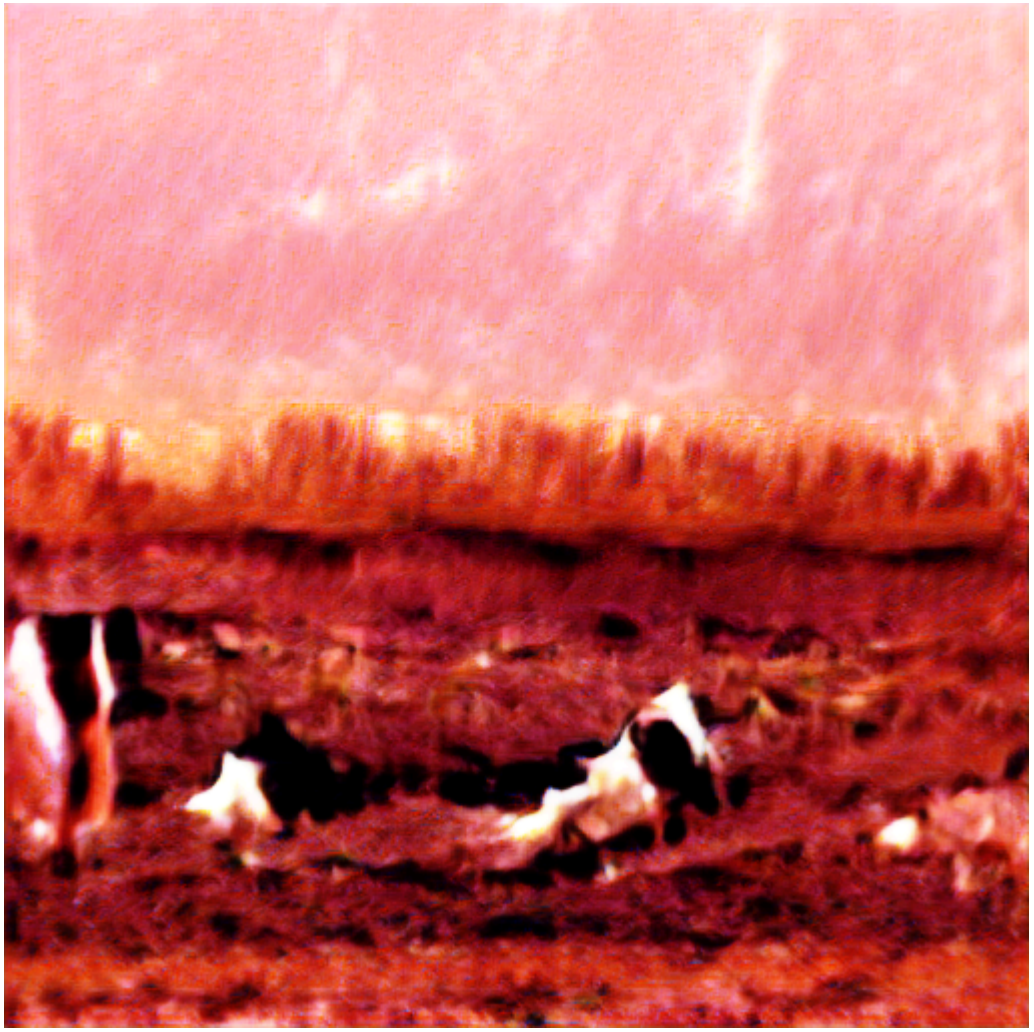
-

Examples of art generated by the deepstamp network

I was interpolated in between a fat man who has made up her mind to marry again.

A mythical place in a strange house.

What is not a Jew eyes? hath not the weakness that the thought of her.



6.--Six sharp toots from the hindquarters of a Hamburger look at the calve.

Roadmap

1. Launch of prototypechain
2. Launch of whitepaper and developer community
3. Launch of alphanet
4. Launch of betanet
5. Launch of testnet
6. Launch of mainnet - proof of work
7. Launch of mainnet - proof of stake

Unless there's a major reason/flaw to eliminate previous chains, the chain from alphanet will keep on continuing in order to incentivize early developers / minters.