Exploiting Computation Power of Blockchain for Biomedical Image Segmentation

Boyang Li, Changhao Chenli, Xiaowei Xu, Taeho Jung, Yiyu Shi
University of Notre Dame, Notre Dame, Indiana USA
{bli1,cchenli,xxu8,tjung,yshi4}@nd.edu

Abstract

Biomedical image segmentation based on Deep neural network (DNN) is a promising approach that assists clinical diagnosis. This approach demands enormous computation power because these DNN models are complicated, and the size of the training data is usually very huge. As blockchain technology based on Proof-of-Work (PoW) has been widely used, an immense amount of computation power is consumed to maintain the PoW consensus. In this paper, we propose a design to exploit the computation power of blockchain miners for biomedical image segmentation, which lets miners perform image segmentation as the Proof-of-Useful-Work (PoUW) instead of calculating useless hash values. This work distinguishes itself from other PoUW by addressing various limitations of related others. As the overhead evaluation shown in Section 5 indicates, for U-net and FCN, the average overhead of digital signature is 1.25 seconds and 0.98 seconds, respectively, and the average overhead of network is 3.77 seconds and 3.01 seconds, respectively. These quantitative experiment results prove that the overhead of the digital signature and network is small and comparable to other existing PoUW designs.

1. Introduction

Deep learning plays a crucial role in supporting clinic diagnosis, especially the biomedical image segmentation algorithm [29][36][30][35][33][18][12] which has been successfully applied in brain MRI segmentation [25], lung CT scans segmentation [14] and Cardiac MRI Segmentation [1]. Instead of segmenting images manually, these deep learning based algorithms accelerate medical diagnosis in terms of extracting different tissues, organs, pathologies, and biological structures. However, it is extremely challenging due to high variability in medical images, low contrast, and other imaging artifacts [36]. By taking advantage of rapid increase of computing power and machine learning research interests [32][31][37][38][34][17][16], segmentation based on deep learning is much less expensive and time-consuming than manual segmentation. However, biomedical image segmentation does require high computation power. In addition, an accurate model requires machine learning experts to tune it by training and evaluating with different hyper-parameters multiple times. Therefore, a good model comes at the cost of very high computation power.

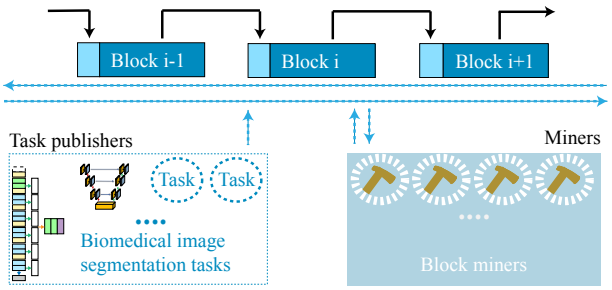

Figure 1. A blockchain maintained with image Segmentation algorithm as PoUW.

Bitcoin [22] is the most popular blockchain technology-based application. Besides countless cryptocurrencies, blockchain technology has been successfully applied in different fields. However, the traditional consensus mechanism demands an immense amount of energy for computation to maintain the blockchain. According to Digiconomist [10], the estimated power consumption of Bitcoin "mining" reaches around 70 TWh per year during the second half of the year 2018. As a result, there are the concerns and warnings about energy wasting of cryptocurrencies [9], for instance, Camilo Mora published a paper in Nature Climate Change with the title of "Bitcoin emissions alone could push global warming above 2 centigrade" [21].

To maintain the consistency of transactions, the traditional proof of work (PoW) consensus mechanism utilizes the brute-force algorithm to host a competition of hardware and energy source, and this is the major component that

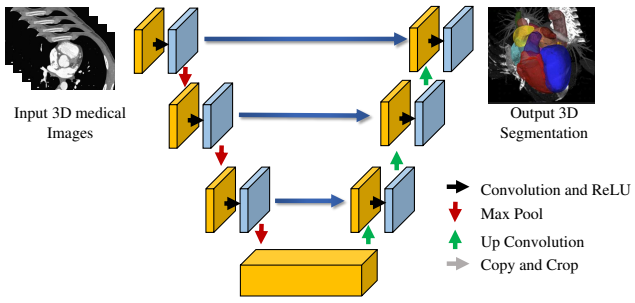

Figure 2. 3D U-Net [11]: a widely used framework in fully convolutional networks for medical image segmentation.

leads to the energy wasting issue. A series of solutions have been proposed to address this issue, such as ASIC machine [23], proof of capacity (PoC) [4] and proof of useful work (PoUW) [23]. ASIC machines compute hash efficiently, but this type of machine is only able to calculate on a certain type of brute-force algorithms and it is relatively inflexible. PoC significantly wastes disk space instead of electricity. On contrary, PoUW exploits computation power of "miners" for useful tasks, therefore the energy consumed by the miners is not wasted.

Primecoin [15], PoX [28] and PoDL [7] are PoUW mechanisms that ask miners to perform useful work. This paper proposes a practical PoUW mechanism to exploit the computation power of "miners" for biomedical image segmentation tasks while addressing the limitation of existing PoUW mechanisms. Our major contributions are: (1) the useful work in our mechanism has a clear useful application; (2) our mechanism can accept and handle multiple tasks; (3) our mechanism can handle larger models and training dataset.

2. Background and related work

Biomedical image segmentation: Fully convolutional networks (FCN) is a special category of DNN, which is widely used for medical image segmentation. Compared with general DNNs, FCNs only have convolutional layers, up convolutional layer, and pooling layers as shown in Fig. 2. With this characteristics, FCNs can efficiently output images with the same size as the input images as shown in Fig. 2. Almost all the DNN-based methods for 3D image segmentation adopt FCN as the backbone network structure, and add some special structures and improved training strategies [6] [8] [20] [5] [11]. For example, 3D U-Net [8] adds more connections between the first several layers and the last several layers as shown in Fig. 2 to better extract features.

Consensus mechanisms for blockchain: Proof of Stake (PoS) is a consensus mechanism in cryptocurrencies to decide the creator of the next block based on the amount of cryptocurrencies the creator own or other weights that can prove the authority of the creator. Determination of the block creators involves efficient computation only, and it does not consume much energy. However, it is unknown whether PoS mechanism is a robust distributed consensus mechanism owing to various limitations [26] [24] [27]. Existing cryptocurrencies adopting PoS all have extra rules to make the consensus mechanism more robust, however, the necessity of extra rules implies the PoS is inherently unstable, and the benefit of energy efficiency is diluted.

Proof of Work (PoW) does not suffer from the instability of PoS. Its stability is supported by the enormous computation resources contributed to hard computation problems, and its criticism has been on its large amount of wasted energy. The idea of PoW was proposed to prevent from the denial-of-service attacking [2]. In PoW consensus, all participants are required to solve problems before they send messages. Such problems are challenging to solve and easy to validate. In Bitcoin, miner nodes are required to calculate hash values as PoW.

Proof-of-Deep-Learning (PoDL) In this paper, we inherit the block acceptance policy of PoDL to substitute the hash calculation with segmentation model training. We briefly explain the mechanism before introducing our novel protocols that address PoDL's limitations.

Each block interval is divided into two phases, Phase 1 and Phase 2. At the beginning of Phase 1, a task publisher releases the training dataset (with labels) as well as the hyperparameters of the deep learning model to miners and full nodes. During Phase 1, miners train their own deep learning model, and commit their model by the end of Phase 1. The commit process is completed through submitting the hash of their trained model as well as their ID. At the beginning of Phase 2, the task publisher releases the test dataset to miners and full nodes, and each miner submits (1) the block header and the block that contains information describing the trained model on top of existing attributes, (2) the trained model, and (3) the accuracy of the trained model, to full nodes. Note that the hash of the block header does not need to be smaller than the threshold because the hard computation is replaced with the model training. Full nodes, during Phase 2, validate the submitted models to check whether they have the claimed accuracy, and this happens on top of existing validation in the blockchain (*e.g.,* validation of the correctness of transactions, merkle tree, hash). To avoid miners over-fitting their models on the disclosed test dataset or stealing others models (published during Phase 2), full nodes discard any block whose model was not committed during Phase 1 (*i.e.,* hash of the model and ID have not been received in Phase 1). Full nodes will accept the block that is submitted with the highest-accuracy model that claimed its accuracy correctly. They choose and validate the models in decreasing order of the claimed ac-

curacy for that.

Such a block acceptance policy yields a robust consensus and is secure against double-spending attack as long as no more than 51% of computation power is owned by the attackers.

However, the PoDL is limited in that they can only handle one task at once, and there is insufficient details about how to handle training model.

To address these drawbacks, we present an alternative PoW mechanism that asks miners to perform biomedical image segmentation tasks and present a trained segmentation model as the proof. The major contribution of this paper are: (1) our blockchain allows submissions of multiple tasks, (2) our blockchain can handle large models with large training data. These contributions are significant since they make the idea of Proof-of-Useful-Work behind the PoDL more practical and applicable in the real world by supporting multiple tasks and larger predictive models.

Other Proof-of-Useful-Work mechanisms:

Primecoin [15] is an altcoin that asks the miners to find a special sequence of prime numbers (Cunningham chain) instead. Although the outcome of miners' computation has mathematical and research meaning, *i.e.,* discovering the Cunningham chain. The application of Cunningham chain in the real world is unclear.

Proof of Exercise (PoX) is a design proposed in [28], which is another PoUW mechanism that lets miners perform certain exercises and present the outcome as proof. In PoX, *employers* publish their tasks onto a board and the miners will randomly fetch tasks from it. The limitation of PoX is that they rely on this centralized board that is maintained by a third party, which significantly dilutes the decentralization property of the blockchain.

3. Definitions and assumptions

In this section, we define the entities involved in our blockchain which will be used to support biomedical image segmentation.

Miners are the machines of individual or small organizations who wish to contribute their computation power for maintaining a blockchain and may receive rewards as the exchange. In our case, we only consider a standard computer (not ASIC machine) with one or more dedicated graphic cards as a miner, for instance, the gaming machines and deep learning machines. Miners train the DL models as PoW with GPUs, maintain a max heap of submitted tasks based on task rewards and validate the result of potential block owner.

Full nodes record all blocks and transactions, maintain a min heap of submitted tasks based on task reward, validate the submitted task and check the checkpoint of miners and validate the result of potential block owner.

Task publishers release biomedical image segmentation training tasks and training data. After a training task is selected and performed by the miners, the corresponding publisher will pay certain amount of reward to the miner presenting the best image segmentation model in the form of the cryptocurrency that is maintained by the blockchain.

There are three assumptions that our design relies on. Some of them hold naturally in existing blockchains while others do not.

Assumption 1: We assume task publishers' best interest is to achieve the image segmentation model with the best performance. Therefore, we assume no collusion happens between miners and task publishers, because colluding with miners (*e.g.,* disclosing test datasets to specific miners) will degrade the accuracy of the model only. However, it is true that miners are well motivated to collude with task publishers (even though task publishers are not motivated to do so) since winning miners gain block rewards. It is our future work to achieve a robust consensus mechanism that does not rely on this assumption.

Besides, we also assume task publishers will pay the task reward honestly once their tasks are performed by the miners. However we introduce how to relax this assumption via smart contract by the end of this paper.

Assumption 2: We assume the training tasks can be interrupted and stopped at any time by the miners. We make this assumption because training tasks may be complex and time consuming, but we need to guarantee certain block generation rate. The gap between the length of training time and the short block interval will be handled by allowing the miners to stop the training tasks at any time and submit the saved checkpoint as their proof of work during the block interval. Note that this assumption holds for optimization algorithms that are based on gradient descendent.

Assumption 3: The full nodes' network condition is stable and reliable enough such that all full nodes have the same view on their memory pool and that miners and task publishers can access such view without significant network delay. In addition, we assume the full nodes' clocks are synchronized up to the difference of 5 seconds. These assumptions are necessary for achieving security properties in our blockchain.

4. Design

We propose to exploit the computation power of blockchain for biomedical image segmentation tasks. Most of the computation power of miners will be spent on training segmentation models instead of calculating useless hash values as in existing PoW mechanisms. Each new block is generated by the miner who submits the best segmentation model, which will be validated by the full nodes. Once the model is confirmed to be the best, the miner will generate the block and receive both of the task reward and block re-

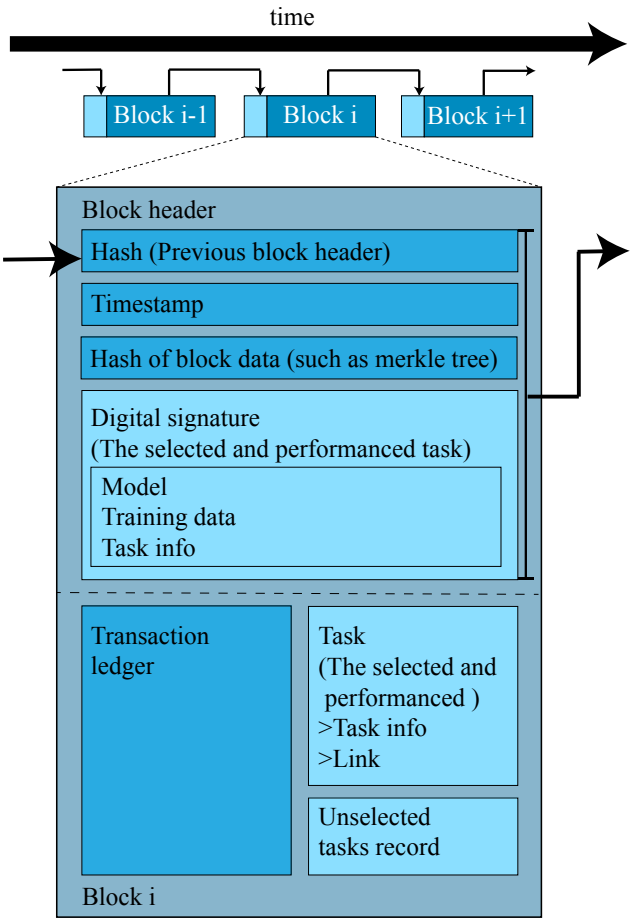

Figure 3. Overviews of the block i.

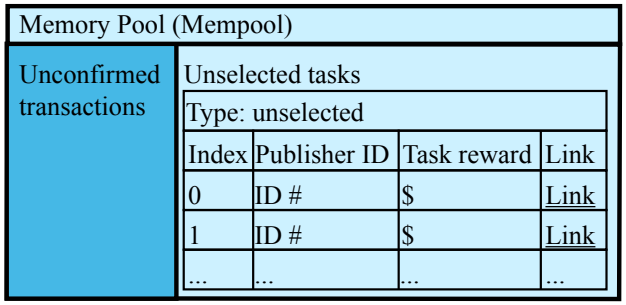

Figure 4. Our augmented memory pool.

ward. The major novelty of this paper is to overcome the limitation of a prior work that cannot handle more than one task and large deep learning model and large training dataset.

In a traditional blockchain, the attributes of the block header, as shown in Fig 3, include the block number, the hash value of the previous block header, the hash of the block data, a timestamp, the size of the block and the nonce value; the block data contains a ledger that records transactions [39]. We introduce three new attributes to the block header in our blockchain: (1) digital signature of segmentation model, training data, and segmentation task information, (2) the task that is selected and performed by all miners, (3) the list of all unfinished tasks that need to be performed in the future. Each attribute will be explained in the subsequent sections.

4.1. Handling multiple tasks

Unlike in [7] where at most one task can be handled by the blockchain, our novel blockchain is capable of handling multiple tasks. We achieve this by augmenting the full nodes' mempool to keep all the unselected tasks (Fig 4). Namely, multiple tasks submitted by the publisher will reside in the mempool until they are selected and performed by the miners.

We firstly define **phase**. As shown in Fig 5, a block interval is split into two parts which are named as Phase 1 and Phase 2, respectively. For block i, Phase 1 starts at time t_{ia} and ends at time t_{ib}; Phase 2 starts at time t_{ib} and ends at time $t_{(i+1)a}$ (*i.e.,* the time when Phase 1 for the next block starts). The period of Phase 1 and Phase 2 are fixed where the length of time for Phase 1 is much longer than that of Phase 2.

We allow the task publishers to submit their tasks to full nodes only during Phase 1. To submit a task, the publisher will need to broadcast the followings to full nodes: publisher ID, task reward value, a link for downloading training dataset as well as the model (*i.e.,* its hyperparameter). At the same time, the publisher will write and launch the smart contract that will send the task reward to the winning miner later when the task is performed and corresponding model is announced in the blockchain. Once the publisher submits a task, it will go to the full nodes' mempool and become an unfinished task. The unfinished tasks will stay in the mempool until the miners select and perform them.

With multiple tasks, it becomes important to let miners agree on the same task to be performed. Otherwise, it is hard to choose the winner by choosing the highest-accuracy model, since comparison of accuracy among different tasks is meaningless. Furthermore, as we will describe in Section 4.4, attackers may attempt to double spend by creating forks, and it is necessary to provide a task selection for the miners to agree on one task for each block.

Our blockchain defines that all miners must choose the task with the highest reward from the unfinished tasks in full nodes' mempool (ties are broken in a pre-defined manner). Due to the assumption that full nodes' views on the mempool are consistent, all full nodes have the same set of tasks in their mempool, and it is the blockchain policy to choose the task with the highest reward, therefore all the miners must select and perform the same task for a specific

block.

4.2. Handling large models and training data

In order to reduce the network traffic, a task publisher will only need to submit the model link and dataset link instead of submitting model and dataset directly during the task submission process. Also, to save the block storage, the link will be stored in blocks instead of actual models or dataset.

Miners still need to retrieve training data from the publisher, which may lead to network delay of tens of seconds or even more. To reduce this time loss, we let task publishers release the training dataset earlier in Phase 2 of the previous block's mining. After Phase 1 for block i, the task to be performed for block $i+1$ has determined already, therefore the miners can start the download. Note that miners are not able to continue training in Phase 2. Otherwise, the model will be different from the one committed in Phase 1. One issue of such training data release is that the miners with high network bandwidth are advantaged because they can start mining earlier than others. To avoid this and make mining fair, we let task publishers encrypt the training data with any efficient symmetric-key encryption (*e.g.,* AES [19]) and release the encrypted training data instead. Then, the publisher releases the key at the end of Phase 2. By doing so, the network delay caused by a key is negligible (*e.g.,* ≤ 256 bits for AES), and the miners who have finished downloading the encrypted training data can decrypt it and perform the training task immediately. The decryption causes extra delay as well, but the decryption itself can be considered as the work that miners need to prove. Note that, with the Assumption 1 in Section 3, the task publisher will not release the key to any specific miners in advance.

Symmetric-key encryption such as AES does not expand data, but it is possible that encrypted training data cannot be fully downloaded within Phase 2 because of the large volume. Motivated miners will monitor the tasks being submitted to full nodes and start fetching the training data even before Phase 2, but our mechanism may have to limit the training data to an acceptable size.

4.3. Our block mining with multiple tasks and large models/data

In this section, the procedure, as shown in Fig. 5, will be introduced in details. In general, the length of Phase 1 is much longer than the length of Phase 2, because the training time will be significantly longer than the testing time. The Phase 1 of block i starts at time t_{ia}. Task publishers can submit image segmentation tasks during this phase. The publisher will need to submit its own ID, reward, and links (model address and data address). In real world scenario that mempool may not hold the same view of task ranking list due to network delay, thus it may need an additional

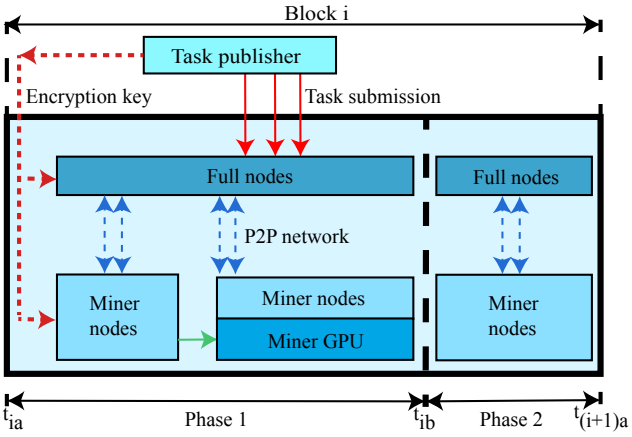

Figure 5. Description of the block mining in detail (block i).

ledger to confirm all submitted tasks and a target task will be selected from confirmed list. However, as it is claimed in the Assumption 3, full nodes will hold the same view on memory pool, thus we won't consider this case and will address this issue in the future. In the current work, all submitted tasks will be recorded in the unselected list, as described in Section 4.1. At the same time, the miners are training the segmentation task which was ranked at the first place by the end of the last block. After a task appeared in the unselected tasks list, it will join the ranking which is based on the task reward. Only the task with the highest task will win the chance to be trained.

Once the target training task is confirmed, miner nodes start fetching data and model, and the publisher of the selected task will release the key to the encrypted model. After the model and data are ready, each miner will evaluate the complexity of the task by training the model for one epoch. Then miners will start training with GPUs for a certain number of epochs. The number of epochs was evaluated by each individual miner and it can be different among miners. This number is measured to ensure that the miner can stop training before Phase 2, yet finish the last entire epoch. The behavior of other participators is described in Section 4.1 4.2.

As shown in Fig 5, the primary job of Phase 2 is to test and validate the biomedical image segmentation model which was trained during Phase 1 of the current block. When the time t_{ib} (shown in Fig. 5) arrives, the publisher will provide API for miner nodes to test their own model and for full nodes to validate the winner model. Miner nodes generate a digital signature which is shown as Fig 3 digital signature frame. At the same time, miners will check the accuracy of their own models. All miners will submit their accuracy values and model links. In addition, the model link is required to include all checkpoint models for verification purpose. This policy is to make sure that the

final model is truly a trained model. The full nodes will sort all submitted accuracy values and verify the model with the highest accuracy. The miner, who submitted the best model, will generate the block. Meanwhile, as described in Section 4.1, the task ledger will be confirmed which also means all unconfirmed tasks are moved into unselected tasks list. The target task for the next block is selected from the updated unselected tasks list, and traditional transaction confirmation is finished.

The essential property of blockchain is that any full node should be able to verify the history data. In this case, it is necessary to check that the accepted biomedical image segmentation models are trained from training dataset only. In addition, the testing results must be the same as it was claimed. As described above, full nodes will be able to fetch all the checkpoint models as the reference to verify the training through the model link in task ledger in block data.

4.4. Handling forks

Instead of considering the longest chain as the correct one, we let full nodes in our blockchain consider the chain who has the most highest-accuracy models as the correct one. The intuition behind this form of fork resolution is similar to that in existing cryptocurrency based on PoW mechanisms. Namely, generating a correct block with a small-enough hash value is challenging in PoW-based cryptocurrency, and a chain will be considered correct if it has the most correct blocks with small-enough hash values (*i.e.,* being the longest chain). In our blockchain, generating a valid block with the highest-accuracy model is challenging, therefore we treat the chain with the most highest-accuracy models as the correct one.

4.5. Validating past blocks

Newly-joined full nodes need to verify the entire blockchain. When checking block i, full nodes will need to check the unselected tasks record from block $i-1$ and the selected task for block i to see whether the task selected in block i has the highest reward. Then, the full nodes will have to verify whether the model accuracy is the same as the one claimed by the winner miner. Then, the full node will verify the digital signature we introduced in the block header to verify the integrity of data. Finally, existing validation (*e.g.,* correctness of hash calculation, transaction validity) will be performed.

4.6. Properties of our blockchain

Synchronized tasks: The augmented mempool stores all unselected tasks, and these mempools will be stored in every block. Miners will have to select the task with the highest reward from this list (that is available in the previous block), therefore all miners are able to agree on the task to perform. Therefore, full nodes are guaranteed to deal with the same task during the block validation. We highlight that this synchronization is achieved without relying on third-party entities, therefore it does not harm the decentralization of blockchain.

Redefining confirmation: Because of the way full nodes choose the next block in our blockchain, whether blocks can be reversed does not depend too much on the number of confirmations (*i.e.,* the number of blocks after them on the blockchain). Rather, the accuracy of the models on the blockchain determines it. Namely, if the block contains a model with a high accuracy, it is challenging to generate another block with another model with a higher accuracy. Then, reversing the previous blocks ahead of the block with a high-accuracy model requires the amount of work needed for training a higher-accuracy model. Therefore, the blocks become hardly reversible after there are multiple high-accuracy models along the chain. Then, we may define the confirmations of a block as the number of high-accuracy models appearing after it rather than the number of blocks after it.

Hardness of double spending: Full nodes will accept the blocks in Phase 2 if and only if their headers are received in Phase 1. Therefore, as long as full nodes are honest, even if adversaries delay the submission of their blocks in order to afford more time in training, they are not allowed to submit blocks with the *better* models (who were trained with more time) because the block headers did not appear in Phase 1.

Even if the majority of the full nodes collude with miners, double spending without 51% computing resources is still a low-probability event. During training process, the optimization algorithms seek local optima with certain randomness because no known algorithms can strategically find the global optimum. Therefore, if only the highest-accuracy models are accepted, it is challenging to further improve the accuracy beyond it, as shown in Fig. 7 9. If adversaries wish to double spend in our blockchain by controlling majority of the full nodes, they must present another chain with more highest-accuracy models. Because the accuracy of the model depends on the hyperparameters and initial weights, choice of which is random, we conjecture that it is extremely difficult to generate another chain with more highest-accuracy models unless the adversary possesses more than 51% of the computing resources for the image segmentation training.

Dataset and model provision: Training dataset may have large volumes, however it is necessary for performing the published tasks. Therefore, we assume the task publishers will host training dataset for the miners' access.

Besides, full nodes who need to verify the whole chain (*e.g.,* newly-joined full nodes) need to access the historical models provided by the winning miners. We also let task publishers store the image segmentation models they col-

lected from the miners, and provide the models to full nodes for their verification. There may be model privacy concerns, however addressing privacy concerns is an orthogonal problem, and we do not address that in this paper.

In case the storage of models (100KB-10GB per model) becomes a burden to the task publisher, we can save the storage by freeing up some earlier models with lower accuracy because the tamper-proofness is guaranteed by high-accuracy models only. Accordingly, we can also let full nodes verify the high-accuracy models only. By doing so, the blocks with high-accuracy models will still prevent the double spending, and the publishers need to store one model (the ultimate one that has the highest accuracy) per task only.

Network delay: Unlike the existing work [7], blocks submitted to full nodes do not include the trained models any more. Instead, the block contains the links providing access to the models, therefore blocks do not need to be very large. Our blockchain does require some extra attributes in the block header as well as various information of tasks in the block. However, the storage burden of those extra data in the block is negligible.

However, miners' access to training data does involve non-negligible network delay which owing to the characteristics of the tasks performed by the miners. If tasks do not need to take large data as input, miners will experience less extra network delay.

5. Experiment

5.1. Experiment setup

The experiments were conducted in small scale local network on the machines with Intel(R) Core(TM) i7-6850K CPU @ 3.60GHz, 32Gb RAM, GTX 1080 Ti.

To exploit computation power of blookchain for the image segmentation tasks, we adopt two widely used networks: fully conventional networks (FCN) and U-net for for 2D and 3D biomedical image segmentation respectively.

For FCN, we adopt the same network as that in the work [40], a 34-layer FCN, which applied bottleneck design and modified the decoding part to improve the accuracy. We use the MICCAI 2015 Gland Challenge dataset which has 85 training 2D images and 80 test 2D images. The loss function, learning rate, regulation parameters, and training epoch are also the same as that in the work [40].

Table 1. Overhead benchmark based on 1000 times testing

Model	Digital signature (s)		Network (s)	
	AVG.	STD.	AVG.	STD.
U-net (270MB)	1.25	0.05	3.77	0.32
FCN (212MB)	0.98	0.04	3.01	0.29

For U-net, we adopt a general configurations: (a) four resolution steps, and each resolution step contains two layers of 3 x 3 x 3 convolutions, rectified linear unit (ReLu), and 2 x 2 x 2 max pooling/up-sampling; (b) the number of filters in higher resolution step doubles that in its lower resolution step, and the initial (lowest) resolution step. We use the CT images in MMWHS 2017 heart segmentation challenge which has 20 training 3D images and 40 test 3D images. The loss function, learning rate, regulation parameters, and training epoch are the same as that in the work [13]. For both the two networks, we use Dice metric for evaluation.

5.2. Benchmark tests

Instead of the brute-force algorithm, the miner nodes performed image segmentation tasks as described in Section 5.1. Fig 6 8 shows the segmentation results with FCN method and U-net method, respectively. The accuracy evaluation results of FCN and U-net are demonstrated in Fig. 7 9. It can be seen that additional training based on a well performed model can hardly improve the performance of the model, thus it will prevent from double spending as the discussion in Section 4.6.

Table 1 shows the extra overhead of digital signature and network. The digital signature was achieved by SHA-

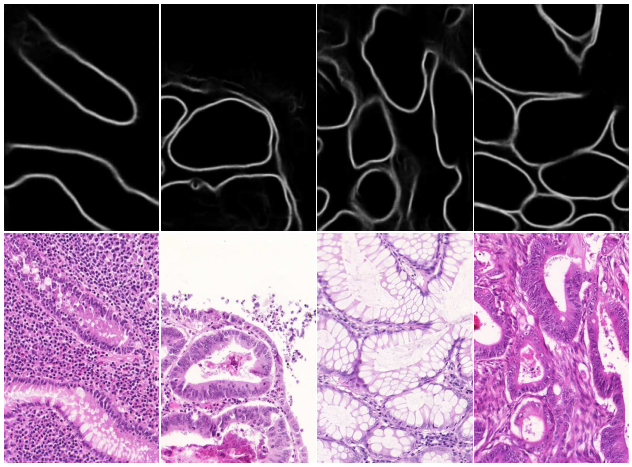

Figure 6. FCN image segmentation result. (The upper/lower row demonstrates the segmentation results and original gland histology images, respectively.)

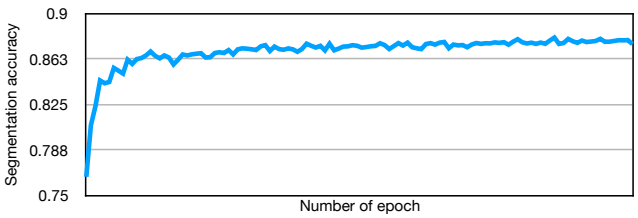

Figure 7. FCN Image segmentation accuracy results.

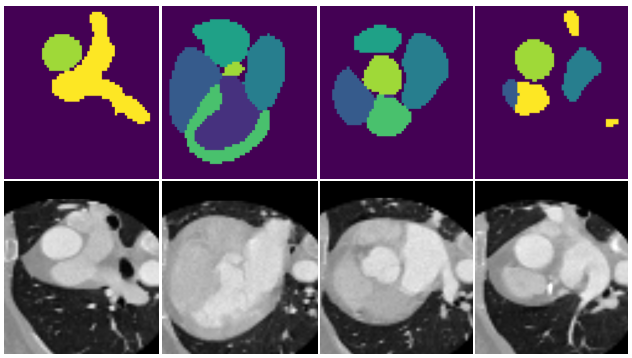

Figure 8. U-net image segmentation result. (The upper/lower row demonstrates the segmentation results and original cardiac CT images, respectively.)

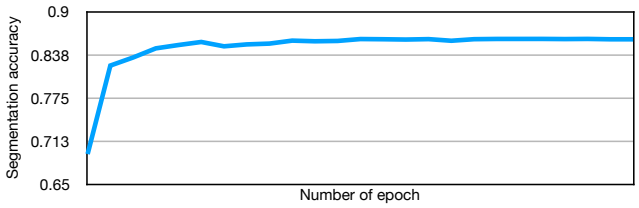

Figure 9. U-net Image segmentation accuracy results.

256 algorithm and the extra network overhead was evaluated by transmitting the winner model through a local network in accuracy validation step. Both overhead are much smaller than image segmentation training time. Therefore, our mechanism utilized most of power on useful tasks and it potentially could be a contribution to both computer vision and blockchain society. Since we assumed the dataset is public accessible, the data loading time is not evaluated in the experiment. Extra storage overhead incurred by augmented memory pool and novel task ledger is negligible which was discussed in Section 4.6.

6. Limitations and future work

In this paper, we presented a blockchain design that lets miners to perform biomedical image segmentation model training instead of hash calculation for block mining. Our blockchain design addresses the limitations of existing PoUW consensus mechanisms. The useful work involved in our design is practical because various disease diagnosis required customized models trained on specific dataset. Our blockchain is able to handle multiple tasks submitted by different task publishers, and it also provide a solution to handle DNN models as well as training dataset with large size. We performed quantitative experiment with real-world data to show that the extra overhead introduced by our design is acceptable.

This work has some limitations at the current version.

```
pragma solidity 0.4.0;
contract TaskContract {
    uint256 private reward;
    uint256 private accuracy;
    string apiForTesting;
    function TaskContract() public{
        taskReward = 1 ether;
        requiredAccuracy = 9500; // 95%
        apiForTesting = xx.yy.com/zz
    }
    function testAndPay(string linkOfModel)
        public{
        require(API_query(apiForTesting,
            linkOfModel) >= accuracy);
        msg.sender.transfer(taskReward);
    }
}
```

Figure 10. A toy example of smart contract that guarantees task reward payment.

We assume full nodes have consistent view as well as synchronized time clock. Achieving a design with the same robustness against various attacks without relying on these assumptions is our immediate future work. Besides, we store all unconfirmed tasks in the block, however the block size is limited to several megabytes. This limits the total number of tasks that can be handled by our blockchain. Breaking this limit is another future work.

Task publishers are assumed to be honest in this paper, however this assumption can be relaxed if we adopt smart contract capable of calling external APIs. For example, if we have a function `API_query(string URL,string link)` that sends the link of a model `link` to a web-based API `URL` and returns its accuracy against the test dataset behind the API `URL`, we can let task publishers submit and deploy a smart contract transaction that looks like Fig. 10. It sends the task reward (1 ether) to the message sender if s/he provides a link of well-trained model that yields a high-enough accuracy ($\geq$ 95%) after the API call to the publisher's API for testing (*e.g.,* xx.yy.com/test). Then, we can let task publishers announce their tasks by deploying smart contract transactions at the blockchain instead of announcing them to full nodes. By doing so, task publishers are unable to reject the task reward payment. Oraclize [3] can be used to implement such external API call in Ethereum-based smart contract transactions, which supports access to any API on the Internet. However, further study needs to be done to understand the security as well as burden to the full nodes, the miners, and even the task publishers.

References

[1] M. Avendi, A. Kheradvar, and H. Jafarkhani. A combined deep-learning and deformable-model approach to fully automatic segmentation of the left ventricle in cardiac mri. *Medical image analysis*, 30:108–119, 2016.

[2] A. Back et al. Hashcash-a denial of service counter-measure. 2002.

[3] T. Bertani. Understanding oracles, 2016.

[4] Burstcoin. Burstcoin @ONLINE. `https://www.burst-coin.org`, March 2019. (accessed: 03.06.2019).

[5] H. Chen, Q. Dou, L. Yu, J. Qin, and P.-A. Heng. Voxresnet: Deep voxelwise residual networks for brain segmentation from 3d mr images. *NeuroImage*, 2017.

[6] H. Chen, X. Qi, J.-Z. Cheng, P.-A. Heng, et al. Deep contextual networks for neuronal structure segmentation. In *AAAI*, pages 1167–1173, 2016.

[7] C. Chenli, B. Li, Y. Shi, and T. Jung. Energy-recycling blockchain with proof-of-deep-learning. In *IEEE International Conference on Blockchain and Cryptocurrency*. IEEE, 2019.

[8] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 424–432. Springer, 2016.

[9] A. De Vries. Bitcoin's growing energy problem. *Joule*, 2(5):801–805, 2018.

[10] Digiconomist. Bitcoin energy consumption index @ONLINE. `https://digiconomist.net/bitcoin-energy-consumption`, March 2019. (accessed: 03.06.2019).

[11] Q. Dou, H. Chen, Y. Jin, L. Yu, J. Qin, and P.-A. Heng. 3d deeply supervised network for automatic liver segmentation from ct volumes. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 149–157. Springer, 2016.

[12] Y. Gao, Y.-F. Li, S. Chandra, L. Khan, and B. Thuraisingham. Towards self-adaptive metric learning on the fly. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), San Francisco, CA, USA, May 13–17, 2019*, 2019.

[13] F. Isensee, J. Petersen, A. Klein, D. Zimmerer, P. F. Jaeger, S. Kohl, J. Wasserthal, G. Koehler, T. Norajitra, S. Wirkert, et al. nnu-net: Self-adapting framework for u-net-based medical image segmentation. *arXiv preprint arXiv:1809.10486*, 2018.

[14] D. Jin, Z. Xu, Y. Tang, A. P. Harrison, and D. J. Mollura. Ct-realistic lung nodule simulation from 3d conditional generative adversarial networks for robust lung segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 732–740. Springer, 2018.

[15] S. King. Primecoin: Cryptocurrency with prime number proof-of-work. *July 7th*, 2013.

[16] Y.-F. Li, Y. Gao, G. Ayoade, H. Tao, L. Khan, and B. Thuraisingham. Multistream classification for cyber threat data with heterogeneous feature space. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), San Francisco, CA, USA, May 13–17, 2019*, 2019.

[17] Z. Liu, S. Luo, X. Xu, Y. Shi, and C. Zhuo. A multi-level-optimization framework for fpga-based cellular neural network implementation. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 14(4):47, 2018.

[18] Z. Liu, X. Xu, T. Liu, Q. Liu, Y. Wang, Y. Shi, W. Wen, M. Huang, H. Yuan, and J. Zhuang. Machine vision guided 3d medical image compression for efficient transmission and accurate segmentation in the clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8300–8308, 2019.

[19] P. Mahajan and A. Sachdeva. A study of encryption algorithms aes, des and rsa for security. *Global Journal of Computer Science and Technology*, 2013.

[20] F. Milletari, N. Navab, and S.-A. Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 565–571. IEEE, 2016.

[21] C. Mora, R. L. Rollins, K. Taladay, M. B. Kantar, M. K. Chock, M. Shimada, and E. C. Franklin. Bitcoin emissions alone could push global warming above 2 c. *Nature Climate Change*, 8(11):931, 2018.

[22] S. Nakamoto et al. Bitcoin: A peer-to-peer electronic cash system. 2008.

[23] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder. *Bitcoin and cryptocurrency technologies: A comprehensive introduction*. Princeton University Press, 2016.

[24] T. Ogawa, H. Kima, and N. Miyaho. Proposal of proof-of-lucky-id (pol) to solve the problems of pow and pos. In *Blockchain*. IEEE, 2018.

[25] S. Pereira, A. Pinto, V. Alves, and C. A. Silva. Brain tumor segmentation using convolutional neural networks in mri images. *IEEE transactions on medical imaging*, 35(5):1240–1251, 2016.

[26] A. Poelstra et al. Distributed consensus from proof of stake is impossible. *URL: https://download. wpsoftware. net/bitcoin/old-pos. pdf*, 2014.

[27] Y. L. Sanket Kanjalkar, Joseph Kuo and A. Miller. I cant believe its not stake! resource exhaustion attacks on pos. In *Financial Cryptography*, 2019.

[28] A. Shoker. Brief announcement: Sustainable blockchains through proof of exercise. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing*, pages 269–271. ACM, 2018.

[29] T. Wang, J. Xiong, X. Xu, and Y. Shi. Scnn: A general distribution based statistical convolutional neural networkwith application to video object detection. In *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI19)*, 2019.

[30] X. Xu, Y. Ding, S. X. Hu, M. Niemier, J. Cong, Y. Hu, and Y. Shi. Scaling for edge inference of deep neural networks. *Nature Electronics*, 1(4):216, 2018.

[31] X. Xu, F. Lin, A. Wang, X. Yao, Q. Lu, W. Xu, Y. Shi, and Y. Hu. Accelerating dynamic time warping with memristor-based customized fabrics. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(4):729–741, 2018.

[32] X. Xu, F. Lin, W. Xu, X. Yao, Y. Shi, D. Zeng, and Y. Hu. Mda: A reconfigurable memristor-based distance accelerator for time series mining on data centers. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2018.

[33] X. Xu, Q. Lu, T. Wang, Y. Hu, C. Zhuo, J. Liu, and Y. Shi. Efficient hardware implementation of cellular neural networks with incremental quantization and early exit. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 14(4):48, 2018.

[34] X. Xu, Q. Lu, T. Wang, Y. Hu, C. Zhuo, J. Liu, and Y. Shi. Efficient hardware implementation of cellular neural networks with incremental quantization and early exit. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 14(4):48, 2018.

[35] X. Xu, Q. Lu, T. Wang, J. Liu, C. Zhuo, X. S. Hu, and Y. Shi. Edge segmentation: Empowering mobile telemedicine with compressed cellular neural networks. In *Proceedings of the 36th International Conference on Computer-Aided Design*, pages 880–887. IEEE Press, 2017.

[36] X. Xu, Q. Lu, L. Yang, S. Hu, D. Chen, Y. Hu, and Y. Shi. Quantization of fully convolutional networks for accurate biomedical image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8300–8308, 2018.

[37] X. Xu, T. Wang, Q. Lu, and Y. Shi. Resource constrained cellular neural networks for real-time obstacle detection using fpgas. In *2018 19th International Symposium on Quality Electronic Design (ISQED)*, pages 437–440. IEEE, 2018.

[38] X. Xu, D. Zeng, W. Xu, Y. Shi, and Y. Hu. An efficient memristor-based distance accelerator for time series data mining on data centers. In *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2017.

[39] D. Yaga, P. Mell, N. Roby, and K. Scarfone. Blockchain technology overview. Technical report, National Institute of Standards and Technology, 2018.

[40] L. Yang, Y. Zhang, J. Chen, S. Zhang, and D. Z. Chen. Suggestive annotation: A deep active learning framework for biomedical image segmentation. In *International conference on medical image computing and computer-assisted intervention*, pages 399–407. Springer, 2017.