

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
ПЕТРА ВЕЛИКОГО»  
ФИЗИКО-МЕХАНИЧЕСКИЙ ИНСТИТУТ  
ВЫСШАЯ ШКОЛА ПРИКЛАДНОЙ МАТЕМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ  
ФИЗИКИ

**Отчет о прохождении преддипломной практики  
на тему: «Нейросетевые и классические методы для задачи автофокуса  
цифрового микроскопа»**

Анищенко Михаила Денисовича, гр. 5030102/00201

**Направление подготовки:** 01.03.02 Прикладная математика и информатика.

**Место прохождения практики:** СПбПУ, ФизМех, ВШПМиВФ.

**Сроки практики:** с 03.05.2024 по 31.05.2024.

**Руководитель практики от ФГАОУ ВО «СПбПУ»:** Чуканов Вячеслав Сергеевич,  
старший преподаватель ВШПМиВФ, .

**Оценка:** отлично

Руководитель практики  
от ФГАОУ ВО «СПбПУ»



В.С. Чуканов

Обучающийся



М.Д. Анищенко

Дата: 31.05.2024

## СОДЕРЖАНИЕ

Введение .....	3
Глава 1. Постановка задачи и обзор решений.....	4
1.1. Существующие классические методы автофокусировки .....	4
1.1.1. Метод анализа резкости.....	4
1.1.2. Фазовый автофокус.....	6
1.1.3. Фазовый сдвиг .....	9
1.2. Нейросетевые методы .....	10
1.2.1. Нейросеть AF-Net .....	10
1.3. Сравнение статистического и нейросетевого метода .....	11
1.4. Выводы .....	12
Глава 2. Разработка нового нейросетевого подхода .....	12
2.1. Основные требования к разрабатываемому подходу .....	13
2.2. Элементы нейронных сетей .....	13
2.2.1. Сверточный слой.....	13
2.2.2. SE блоки.....	15
2.3. Архитектура нейросети .....	16
2.4. Определение направления смещения.....	19
Глава 3. Обучение нейросети .....	21
3.1. Данные для обучения.....	21
3.2. Результаты обучения.....	22
3.3. Метрики .....	24
3.4. Эксперименты .....	24
Заключение .....	25
Список сокращений и условных обозначений .....	27
Список использованных источников.....	28

## ВВЕДЕНИЕ

В цифровой микроскопии применяются различные методы автофокусировки. Однако по-прежнему один из самых распространенных способов – ручная фокусировка оператором микроскопа. Такой способ очевидно медленный и не самый точный. Большинство автоматических методов неустойчивы к шуму, что сильно влияет на конечный результат.

В последнее время разрабатываются все более качественные и универсальные подходы к автофокусировке микроскопов. Также нельзя не заметить активное использование нейросетей в широком спектре задач. Микроскопия не стала исключением. Но некоторые из методов автофокусировки них требуют внедрения в программное обеспечение (ПО) камеры.

Задачи, решаемые в рамках данной работы:

- А. Изучение существующих алгоритмов.
- В. Разработка нейросети, решающей задачу автофокусировки микроскопов.
- С. Проведение экспериментов и тестов.

Целью этой работы стала разработка нового нейросетевого метода автофокусировки, который бы удовлетворял следующим требованиям:

- А. Высокая скорость работы.
- В. Низкое потребление вычислительных ресурсов.
- С. Использование изображений только с камеры (без использования дополнительных датчиков).

## ГЛАВА 1. ПОСТАНОВКА ЗАДАЧИ И ОБЗОР РЕШЕНИЙ

*Задача фокусировки* – нахождение положения линзы или системы линз относительно матрицы камеры, в котором снимок объекта будет максимально четким, то есть наведение объекта схемки на резкость.

*Автофокусировка* – фокусировка, которая осуществляется алгоритмом без участия человека.

*Цифровой микроскоп* – вид оптического (светового) микроскопа, в котором для наблюдения объекта используется цифровая камера вместо окуляров или вместе с ними.

Необходимо разработать новый алгоритм автофокусировки цифрового микроскопа, который отличался бы легкостью применения, высокими скоростью работы и метриками качества даже при наличии шума. Кроме того, алгоритм должен работать исключительно с изображениями с камеры, то есть без применения каких-либо дополнительных датчиков.

### 1.1. Существующие классические методы автофокусировки

Существует множество классических методов автофокусировки. Например, контрастный, фазовый (Phase-Detection Autofocus или PDAF) или аналитический метод на основе оценки резкости изображения. В этом параграфе будут рассмотрены их алгоритмы работы.

#### 1.1.1. Метод анализа резкости

Аналитический метод является наиболее простым и примитивным, но из-за этого наименее эффективным. Данный алгоритм работает на основе расчета резкости изображения. Сама же резкость рассчитывается путем нахождения дисперсии лапласиана в каждой точке снимка.

Лапласиан является мерой второй производной и используется для поиска областей с быстрой сменой яркости. Чем выше дисперсия Лапласиана по всему изображению, тем выше «четкость» этого изображения. Аналитически лапласиан определяется формулой

$$L = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (1.1)$$

Однако при работе с изображением, то есть с матрицей чисел, возникает трудность в применении данной формулы, так как нет явной функции, которая описывала бы сделанный снимок. Чтобы решить эту проблему, необходимо более детально рассмотреть, что из себя представляет лапласиан. Лапласиан есть дивергенция градиента заданной функции:

$$\nabla F = \text{div grad} F \quad (1.2)$$

Приблизительно рассчитать градиент в точке можно с помощью оператора Собеля [8]. Оператор Собеля представляет собой дискретный дифференциальный оператор, вычисляющий приближённое значение градиента яркости изображения. Данный оператор производит свертку исходного изображения с ядрами размера  $3 \times 3$ .

Пусть  $\mathbf{A}$  – исходное изображение,  $\mathbf{G}_x$  и  $\mathbf{G}_y$  – изображения, каждая точка которых является приближительным значением производной по  $x$  и  $y$  соответственно.

$$\mathbf{G}_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{A}, \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A} \quad (1.3)$$

$$\mathbf{G} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2} \quad (1.4)$$

Матрица  $\mathbf{G}$  описывает значение градиента в каждом пикселе. Однако для дальнейшего шага потребуются матрицы  $\mathbf{G}_x$ ,  $\mathbf{G}_y$ . Далее нужно посчитать дивергенцию полученного градиента. Для каждой матрицы градиента необходимо посчитать приближительное значение дивергенции по каждой координате. Для каждого пикселя в матрице вычислим его частную производную. Это можно сделать, вычислив разницу между значением этого пикселя и средним значением его соседей.

$$\text{div} f_{i,j} = f_{i,j} - \frac{f_{i-1,j} + f_{i+1,j} + f_{i,j-1} + f_{i,j+1}}{4} \quad (1.5)$$

Получив две матрицы дивергенции, необходимо их сложить, чтобы получить матрицу лапласиана.

$$L = \nabla F = \text{div} \mathbf{G} = \frac{\partial \mathbf{G}_x}{\partial x} + \frac{\partial \mathbf{G}_y}{\partial y} \quad (1.6)$$

Следующим этапом после вычисления матрицы лапласиана является нахождение выборочной дисперсии значений этой матрицы. Именно это значение и будет мерой четкости/резкости изображения.

Для того, чтобы определить положение фокуса, необходимо проделать упомянутые выше шаги для набора снимков и выбрать тот, который имеет наибольшее по абсолютной величине значение дисперсии лапласиана, так как это будет означать максимальные перепады интенсивности, а значит, наиболее выраженные края и границы объекта съемки.

Очевидно, этот подход имеет много недостатков:

- Малая скорость работы. Алгоритм требует целого набора изображений на вход, на съемку которого затрачивается сравнительно много времени.
- Индивидуальная настройка для каждого объектива/камеры. Необходимо сделать серию снимков, выбрать самый качественный из них и вернуть камеру в это положение. Очевидно, что для объективов с большим коэффициентом увеличения разница положений камеры между двумя соседними снимками должна быть меньше, чем для объективов с малым увеличением. При этом диапазон расстояний для любой конфигурации должен быть примерно одинаковым.
- Невысокая точность. У данного алгоритма она ограничивается шагом сдвига камеры по оси  $z$ , так как истинное положение фокальной плоскости может оказаться между двумя зарегистрированными положениями камеры. Тогда в худшем случае ошибка вычисления составит  $\frac{s}{2}$ , где  $s$  – величина шага по вертикали.

### 1.1.2. Фазовый автофокус

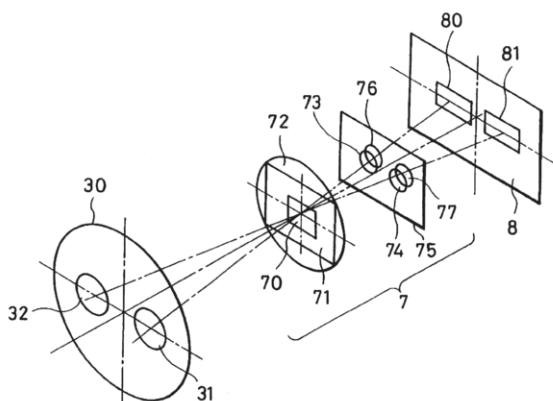


Рис.1.1. Устройство фазового автофокуса. 72 - конденсор; 8 - датчик; 80, 81 - ПЗС-линейки; 30 - зрачок объектива; 31, 32 - области зрачка; 75 - маска; 76, 77 - микрообъективы для датчика; 70 - окно, ограничивающее размер изображения; 71 - рамка конденсора.

Принцип работы фазового автофокуса следующий: в матрицу (рис.1.1) встроен фазовый датчик (8) с двумя линейками (80, 81), в которые приходят световые потоки с противоположных областей (31 и 32) зрачка (30) объектива через микрообъективы (76 и 77). Если объект в фокусе, изображения объекта находятся в центрах соседних ПЗС-линеек. Таким образом, сигналы, получаемые процессором с разных линеек, совпадают (находятся «в фазе»). Если же объектив не сфокусирован на объекте съемки, то изображения смещаются внутрь или наружу линеек. Сигналы перестают совпадать и имеют разность фаз (рис.1.2). На основе этой разности фаз считается фазовый сдвиг, а далее – необходимое смещение для системы линз.

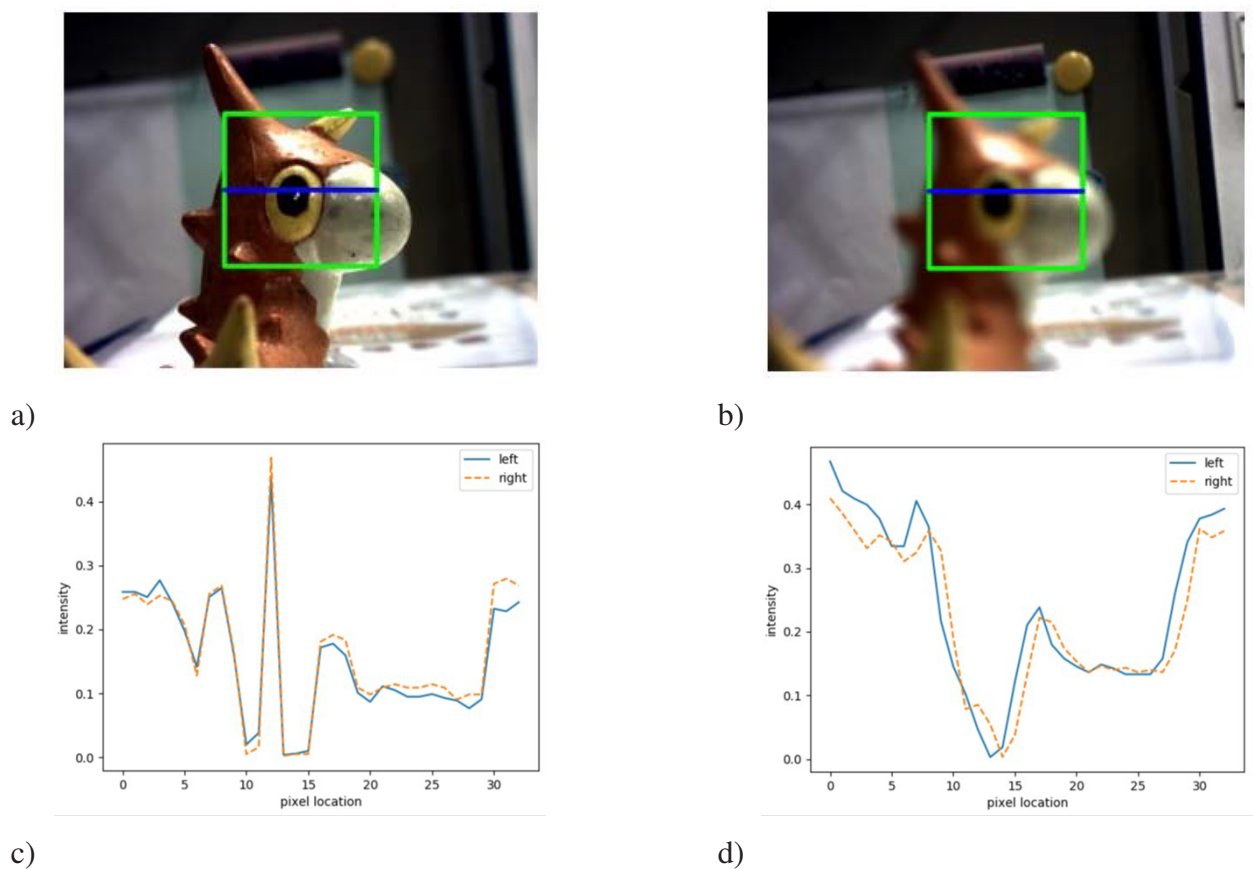


Рис.1.2. Снимки и срезы их фазовых изображений: *a* — снимок объекта в фокусе; *b* — снимок объекта не в фокусе; *c* — срез фазового изображения, когда объект в фокусе; *d* — срез фазового изображения, когда объект не в фокусе

*ПЗС-линейка* – специализированная аналоговая интегральная микросхема, состоящая из светочувствительных фотодиодов, выполненная на основе кремния, использующая технологию ПЗС – приборов с зарядовой связью (полупроводниковых приборов, в которых применяется технология управляемого переноса заряда в объёме полупроводника).

Для PDAF знак фазового сдвига определяет направление движения линзы, а величина фазового сдвига определяет расстояние перемещения линзы. Фазовый сдвиг положителен, если фокальная плоскость находится перед объектом, и отрицателен, если фокальная плоскость находится позади объекта. Чем дальше объект от фокальной плоскости, тем больше абсолютная величина фазового сдвига.

Однако оценка фазового сдвига редко бывает безошибочной. Шум датчика, размытие изображения и низкая текстура часто влияют на точность и надежность оценки фазового сдвига. Хотя ошибку оценки фазового сдвига можно уменьшить, применив фильтр Гаусса к фазовой корреляции, по-прежнему трудно получить точную оценку фазового сдвига, когда объектив находится далеко от сфокусированного положения линзы. Другая заметная проблема PDAF заключается в том, что ошибка оценки фазового сдвига может накапливаться в процессе принятия решения о движении линзы.

Для оценки фазового сдвига в частотной области обычно используется фазовая корреляция. Он применяется для поиска соответствия между смещенными, повернутыми и разномасштабными изображениями. Но этот метод может легко потерпеть неудачу при работе с размытыми или зашумленными изображениями. Чтобы решить эту проблему, было решено применять фильтр Гаусса и использовать разность Гауссианов для извлечения признаков.

В параксиальном и тонколинзовом приближениях формирования изображения глубина объекта  $D$  связана с фазовым сдвигом  $s$  соотношением [3]

$$s = A \left( \frac{1}{z} - \frac{1}{D} \right), \quad (1.7)$$

где  $A$  – некоторая константа,  $z$  – расстояние от фокальной плоскости до линзы. Когда объект находится в фокусе,  $z = D$ . Однако уравнение (1.7) может не выполняться при наличии ошибки оценки фазового сдвига. К этому может приводить ситуация, когда левый и правый фазовые датчики расположены слишком далеко друг от друга.

В качестве решения данной проблемы предлагается использовать статистический метод, который сначала получает распределение вероятностей оптимального расстояния хорда линзы для заданного фазового сдвига на этапе калибровки, а затем использует его для определения расстояния хода линзы при тестировании.



### 1.1.3. Фазовый сдвиг

Для идеальной пары левого и правого фазового изображения с фазовым сдвигом  $\Delta x$  выполняется следующее:

$$r(x, y) = l(x + \Delta x, y), \quad (1.8)$$

где  $(x, y)$  – координаты пикселя,  $r(\cdot, \cdot)$  и  $l(\cdot, \cdot)$  – правые и левое фазовые изображения соответственно.

Фазовый сдвиг между левым и правым изображениями можно получить с помощью фазовой корреляции [10]. Пусть  $L$  и  $R$  – двумерное преобразование Фурье для левого и правого фазовых изображений соответственно. Для этого найдем корреляционную матрицу  $p(x, y)$ .

$$p(x, y) = F^{-1} \left\{ \frac{L \circ \bar{R}}{|L \circ R|} \right\} \quad (1.9)$$

где  $F^{-1}$  – обратное двумерное преобразование Фурье, « $\circ$ » – поэлементное умножение матриц, « $\bar{\cdot}$ » – комплексное сопряжение.

Далее рассматриваем корреляционную прямую  $p(x, 0)$ . Аргумент  $y$  принят равным нулю, поскольку матрицу значение  $p(x, y)$  соответствует корреляции при сдвиге на  $x$  и  $y$  по соответствующим осям. В данном случае сдвиг производится лишь по одной оси, поэтому второй аргумент равен нулю. Однако шум от датчиков может приводить к нескольким пикам в корреляционной матрице. Чтобы этого избежать, применим сглаживание корреляционной кривой  $p(x, 0)$  Гауссовым ядром  $g(x)$ :

$$p_f(x) = g(x) * p(x, 0), \quad (1.10)$$

где  $p_f(x)$  – исправленная корреляционная кривая. Тогда фазовый сдвиг  $\Delta x$  определяется как

$$\Delta x = \arg \max_x p_f(x) \quad (1.11)$$

Если построить график фазового сдвига в зависимости от положения линзы, полученная кривая будет называться профилем фазового сдвига. Положение линзы, соответствующее нулевому фазовому сдвигу, является положением линзы в фокусе. Аналогично, если мы построим график контрастности изображения в зависимости от положения линзы, получим профиль фокуса (рис.1.3). На нем положение линзы в фокусе соответствует максимальному значению контрастности и оно совпадает с нулевым фазовым сдвигом.

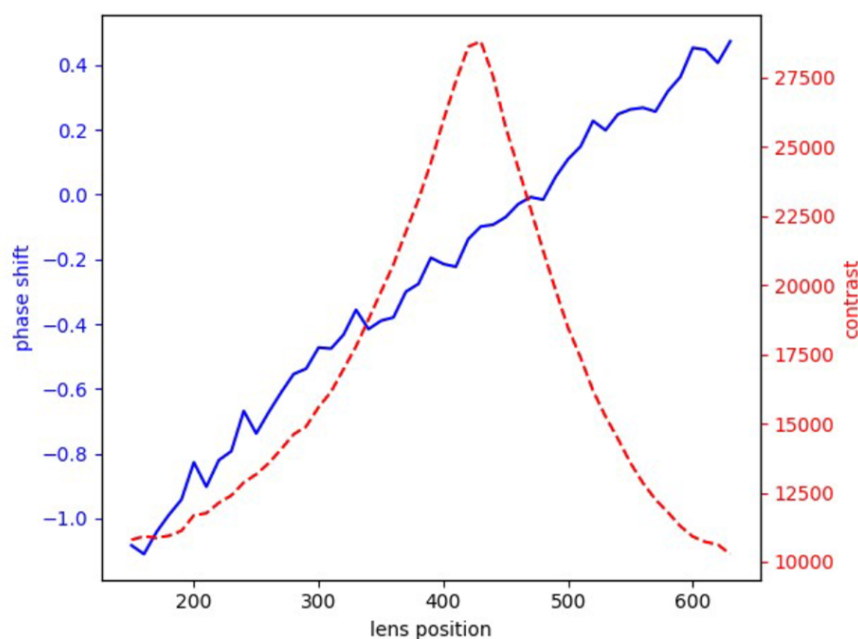


Рис.1.3. Фазовый профиль и профиль фокуса

## 1.2. Нейросетевые методы

Фазовый автофокус показывал отличные результаты по сравнению с более старыми методами автофокусировки. Но упомянутые выше недостатки, в частности неустойчивость к шуму, побуждали искать новые методы работы с оптикой и камерами. Кроме того с ростом интереса к нейросетям и машинному обучению были предприняты попытки использовать их и в этой области. Было предложено несколько решений, использующих глубокое обучение. Некоторые из них определяли смещение камеры на основе разности двух изображений [1], другие по двум снимкам генерировали виртуальный сфокусированный снимок [6]. Однако для уменьшения временных затрат на фокусировку исследования также были направлены на разработку методов, которые не требуют дополнительных перемещений камеры. Так была разработана сверточная нейронная сеть на основе алгоритма фазового автофокуса AF-Net.

### 1.2.1. Нейросеть AF-Net

Подход на основе глубокого обучения [3] позволяет уменьшить влияние шума и ошибку, которая может накапливаться при выполнении большого числа операций с фазовыми изображениями.

На вход нейросети поступают те же фазовые изображения (левое и правое), на выходе получается вещественное число со знаком, которое обозначает направление

и величину смещения линзы для достижения состояния максимальной четкости. Архитектура сети выглядит следующим образом:

Выделение признаков				
Названия слоя	Ядро	Шаг ядра	входные/выходные каналы	входной/выходной размер
conv0	$5 \times 5$	$2 \times 2$	2/64	$33 \times 33 / 16 \times 16$
conv1	$3 \times 3$	$1 \times 1$	64/128	$16 \times 16 / 16 \times 16$
conv2	$3 \times 3$	$2 \times 2$	128/128	$16 \times 16 / 8 \times 8$
conv2	$3 \times 3$	$2 \times 2$	128/256	$8 \times 8 / 4 \times 4$
Полносвязные слои				
Название слоя	Входные/выходные размеры		Функция активации	
fc0	4096/256		ReLU	
fc1	256/64		ReLU	
fc2	64/1		Линейная	

Таблица 1.1

### Архитектура нейросети

После каждого сверточного слоя следует функция активации ReLU и пакетная нормализация (batch normalization). Используется оптимизатор Adamax, размер батча равен 128, параметры оптимизатора  $\beta_1 = 0.5$ ,  $\beta_2 = 0.999$ , learning rate равен 0.001. Также процесс обучения было принято останавливать, когда потери не уменьшаются в течение 20 эпох. В среднем процесс обучения задействует 80 эпох.

### 1.3. Сравнение статистического и нейросетевого метода

Сравним AF-Net с классическим методом фазового автофокуса, используя следующие метрики: коэффициент успеха (success rate), количество перемещений линзы до остановки процесса и ошибка конечного положения линзы. Сравним на разных начальных положениях линзы. Обозначим нулем положение линзы в фокусе, положительное смещение - это перемещение линзы ближе к объекту, отрицательное - дальше от объекта, чем положение в фокусе. Результаты:

Также сравним их производительность:

Метод	Метрика	начальное положение линзы					
		-30	-20	-10	10	20	30
AF-Net	Success rate	90.48	94.83	95.06	97.26	100.0	100.0
	final lens position error	1.381	1.207	1.185	1.027	0.981	0.826
	number of lens movements	2.26	2.20	2.03	2.06	2.06	2.00
Статистический	Success rate	23.81	46.55	51.85	63.01	57.41	43.48
	final lens position error	10.57	5.741	4.000	3.342	3.426	3.261
	number of lens movements	3.40	3.00	2.21	2.21	2.54	2.60

Таблица 1.2

Сравнение метрик AF-Net и статистического метода

Метод	Максимальная ошибка положения линзы	Максимальное количество перемещений линзы	Среднее время расчета одного перемещения (мс)
AF-Net	5	4	10.8
Статистический	12	7	28.3

Таблица 1.3

Оценка производительности

## 1.4. Выводы

Из сводных сравнительных таблиц хорошо видно, что нейросетевой подход превосходит классический как в метриках качества, так и по производительности. Это означает, что применение AF-Net оправдано, и благодаря ему, фокусировка будет проходить быстрее и точнее. Но автофокусировка, основанная на фазовых изображениях и фазовом сдвиге, имеет существенный недостаток – она требует интеграции непосредственно в камеру, так как предполагается, что такой алгоритм является частью ПО камеры. Он использует показания фазовых датчиков. Но зачастую подключиться к ним извне нет возможности, или же эти датчики отсутствуют вовсе.

## ГЛАВА 2. РАЗРАБОТКА НОВОГО НЕЙРОСЕТЕВОГО ПОДХОДА

Существующие решения, решающие задачу автофокусировки микроскопа, показывают хорошие результаты, но их главный минус в том, что они не являются универсальными. В этой главе будет рассказано о новом подходе на основе машинного обучения, который не требует никаких особых датчиков и интеграции в ПО камеры.

## 2.1. Основные требования к разрабатываемому подходу

Сложность развертывания и применения решений на основе фазовых изображений побудила к разработке нового алгоритма, который не имеет такого количества ограничений и недостатков. Таким образом получаем следующие основные пункты, которые нужно учесть при разработке:

1. На вход должно подаваться изображение с камеры. Алгоритм не должен задействовать какие-либо датчики системы регистрации, которые могут отсутствовать на большинстве видов аппаратуры. Поэтому самым удобным и очевидным входным параметром является изображения.
2. Алгоритм должен принимать на вход ровно один снимок. Если для расчета фокальной плоскости будет использовать набор изображений, то алгоритм будет работать недостаточно быстро, так как скорее всего потребуется многократное смещение камеры в некоторые промежуточные позиции.
3. Алгоритм должен определять не только величину смещения линзы/камеры, но и его направление. Это также необходимо для достижения максимальной скорости работы.

## 2.2. Элементы нейронных сетей

В данной работе ключевое место имеет подход к задаче автофокуса на основе глубокого обучения. Нейронные сети, которые позволяют работать с изображениями и извлекать различные важные признаки из них, имеют ряд конструктивных особенностей. Далее подробно будут рассмотрены элементы, которые будут применены в разработанном нейросетевом методе автофокусировки и требуют особого внимания.

### 2.2.1. Сверточный слой

*Сверточный слой* – это слой нейронной сети, который выполняет одноименную математическую операцию [2]. Свертка двух функций можно интерпретировать как меру схожести или корреляции этих двух функций. Формально операция свертки определяется следующим образом:

$$s(x) = (f * g)(x) \stackrel{\text{def}}{=} \int_{\mathbb{R}^n} f(y)g(x - y)dy = \int_{\mathbb{R}^n} f(x - y)g(y)dy, \quad (2.1)$$

где  $f, g : \mathbb{R}^n \rightarrow \mathbb{R}$  – функции, интегрируемые в смысле Лебега,  $s : \mathbb{R}^n \rightarrow \mathbb{R}$ .

В случае работы с изображениями вместо функций  $f$  и  $g$  используются многомерные массивы, как правило, двумерные целочисленные. Исходя из этого, стоит переопределить операцию свертки для дискретизированного набора данных следующим образом:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i-m, j-n) K(m,n), \quad (2.2)$$

где  $I$  – исходное изображение,  $K$  – ядро свертки,  $S$  – карта признаков или карта активации,  $(i,j)$  – её пиксель,  $(m,n)$  – индексы, перебирающие пиксели ядра свертки.

В контексте нейросетей и глубокого обучения вместо операции свертки обычно используется родственная ей операция, которая называется *кросс-корреляцией*. Она симметрична функции свертки и имеет тот же смысл, однако перекрестная корреляция проще и быстрее реализуется, так как не требует отражения ядра свертки по обеим осям. Кросс-корреляция имеет следующий вид:

$$(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n) K(m,n), \quad (2.3)$$

Если изображение, подаваемое на вход, является многоканальным (например, формата RGB). В таком случае для каждого канала изображения используется свое ядро свертки. Набор таких ядер называется *фильтром свертки*. При этом таких фильтров может быть несколько. Каждое ядро свертки является обучаемым.

*Свертка по глубине* или *depthwise свертка* [5]. Основная особенность данного типа свертки – это значительное снижение вычислительной нагрузки, а выполняется он в два этапа. На первом этапе, в отличие от стандартной свертки, в свертке по глубине для каждого канала входного тензора используется свое ядро свертки. Таким образом, фильтр depthwise свертки для цветного трехканального изображения будет состоять из трех ядер и на выходе после первого шага будет также трехканальный тензор. На втором этапе применяется привычная свертка с ядром размера  $1 \times 1$  и с указанным числом фильтров, чтобы собрать полученную информацию по признакам в единое целое.

Благодаря такому подходу, вычислительная сложность алгоритма и количество обучаемых параметров значительно уменьшаются, что очень важно, когда алгоритм применяется в условиях ограниченности времени или мощностей [5, с. 2—3].

### 2.2.2. SE блоки

*Squeeze-and-Excitation (SE) блок* – это архитектурное дополнение к сверточным слоям, главная цель которого – улучшение производительности путем отбора признаков, наиболее сильно влияющих на результат. SE блоки перенастраивают веса, усиливая значимые признаки и подавляя менее важные [4]. В предлагаемой нейросети SE блок имеет структуру, изображенную на рис.2.1. Такой блок обрабатывает данные в три шага:

1. Сжатие (Squeeze). На это этапе происходит глобальное объединение по среднему (global average pooling) на каждом канале входного изображения, сжимая пространственные показатели признаков в одно число. Это позволяет извлечь глобальную информацию с о канале.
2. Возбуждение (Excitation). Этот шаг объединяет в себе несколько слоев. Сначала используется полносвязный слой с функцией активации ReLU, зачет еще один полносвязный слой с логистической функцией активации (сигмоидой). Первый слой уменьшает количество каналов с  $C$  до  $C/r$ , где  $r$  – некоторый коэффициент. В большинстве случаев его принято устанавливать равным 16. Второй слой восстанавливает исходное количество каналов. Такой процесс позволяет улавливать и моделировать взаимосвязи между каналами. Кроме того, механизм понижения и восстановления размерности действует как регуляризация и уменьшает возможность переобучения. Стоит отметить, что на практике (например, в библиотеке Pytorch) чаще встречаются реализации SE блоков с использованием свертки с ядром  $1 \times 1$  вместо полносвязных слоев. Это обусловлено тем, что в таком случае сохраняется пространственная структура, что позволяет более естественно и удобно работать с многомерными данными, коими являются изображения и результаты применения сверточных слоев. То есть не приходится выравнивать многомерный массив (такая операция называется *flattening*). А также применения свертки более эффективно с точки зрения вычислений и памяти, так как графические и тензорные процессоры оптимизированы для выполнения именно этой операции, и их ресурсы будут расходоваться экономичнее.
3. Перенастройка весов (Reweight). Весовые коэффициенты, полученные на прошлом этапе, поэлементно умножаются на исходные признаки.



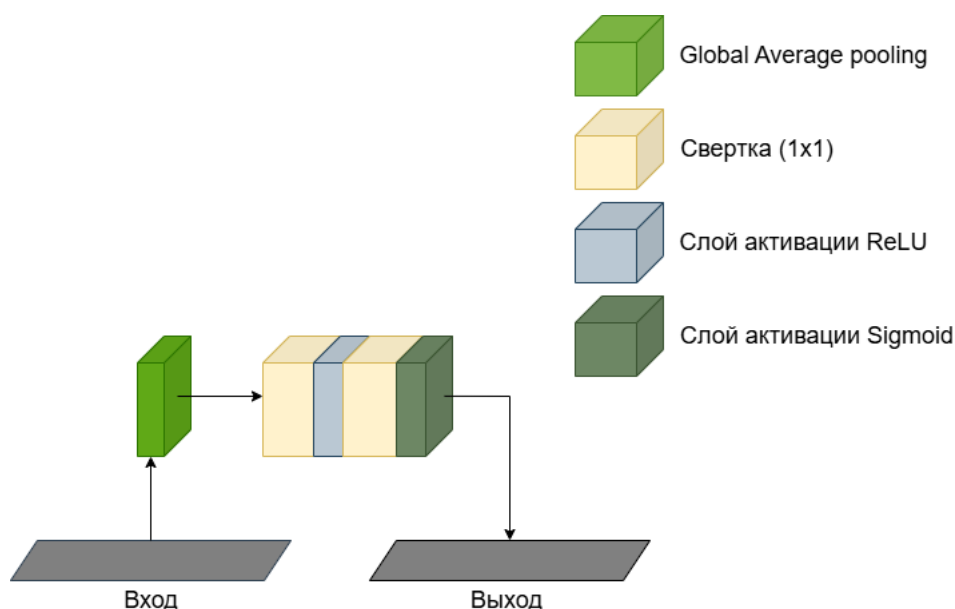


Рис.2.1. Squeeze-and-Excitation блок

### 2.3. Архитектура нейросети

В основе разработанного решения лежит сверточная нейросеть MobileNet от Google, которая была разработана для классификации объектов на изображении на смартфонах. Существует несколько модификаций данной нейросети. Для решения задачи автофокуса была выбрана версия MobileNetV3\_Small. Одним из ключевых факторов в принятии решения была скорость работы и легковесность, так как принимать решение о смене позиции камеры нужно максимально быстро, не затрачивая больших вычислительных ресурсов.

Главная идея данной сети – так называемые bottleneck-блоки. Они используют depthwise свертку, преимущество которой заключается в снижении вычислительной нагрузки благодаря меньшему количеству операций и меньшему числу обучаемых параметров. В следствие этого уменьшается время обработки входных данных нейросетью, а сам алгоритм становится менее требователен к оборудованию в условиях ограниченных мощности и памяти.

Далее были внесены некоторые изменения в архитектуру MobileNet, чтобы адаптировать ее для решения поставленной задачи. Изменилась сама цель алгоритма: если оригинальная сеть предназначалась для классификации, то модифицированная версия для автофокуса должна решать задачу регрессии, так как ее цель – предсказание непрерывной величины, а именно смещения камеры. В данном случае набором признаков (или факторов) для предсказания данной величины является само входное изображение.



Для того, чтобы адаптировать алгоритм под задачу регрессии, необходимо было внести изменения в последние слои, заменив классификатор на последовательность слоев, выполняющих регрессию. Классификатор состоял из двух полносвязных слоев, функции активации *hardswish*, слоя исключения (далее *dropout* или *дропаут*). Регрессор же состоит из двух полносвязных слоев и функции активации *ReLU*.

В качестве функции потерь была выбрана *smooth L1-loss*, которая является комбинацией *L1-loss*, также известной как *Mean Absolute Error (MAE)* или средняя абсолютная ошибка, и *L2-loss* – *Mean Squared Error (MSE)* или средней-квадратичной ошибкой. *Smooth L1-loss* задается следующей формулой:

$$L1_{smooth} = \begin{cases} \frac{1}{2\beta}(y - y')^2, & |y - y'| < \beta \\ |y - y'| - \frac{\beta}{2}, & \text{иначе} \end{cases}, \quad (2.4)$$

где  $y$  – истинное искомое значение,  $y'$  – предсказанное значение,  $\beta$  – некоторая константа, которая задается при настройке сети и в данном случае равная единице. Выбранная функция потерь имеет следующие преимущества:

1. Стабильность градиента. *Smooth L1-loss* малых ошибок ведет себя как *MSE*, а для больших – как *MAE*. Константа  $\beta$  задает величину ошибки, на которой происходит это разделение. Но как известно, градиент функции *L1-loss* не определен в нуле, из-за чего могут возникать трудности при обучении во время обратного распространения ошибки. Вероятность совпадения истинного и предсказанного значений мала, но все же такое случается. Градиент функции средне-квадратичной ошибки лишен этого недостатка, но эта функция потерь вносит слишком большое влияние при больших ошибках. Градиент функции *smooth L1-loss* выглядит следующим образом:

$$\nabla L1_{smooth} = \begin{cases} \frac{y - y'}{\beta}, & |y - y'| < \beta \\ \text{sign}(y - y'), & \text{иначе} \end{cases} \quad (2.5)$$

Это способствует лучшей сходимости алгоритма обучения.

2. Устойчивость к выбросам. Функция потерь *MSE* чувствительна к выбросам из-за квадратичной зависимости от ошибки. *Smooth L1* при больших отклонениях ведет себя линейно, что позволяет избежать слишком большого влияния ошибок.

Таким образом, Smooth L1-loss является некоторым симбиозом функций абсолютной и среднеквадратичной ошибки, вобрав в себя лучшее от каждой из них.

Описание архитектуры нейросети представлено в табл.2.1. Более детальную схему слоев можно увидеть на рис.2.2.

Вход	Слой	exp size	Выход	SE	AF	stride
$672^2 \times 3$	conv2d, $3 \times 3$	-	16	-	Hardswish	2
$336^2 \times 16$	bneck, $3 \times 3$	16	16	✓	ReLU	2
$168^2 \times 16$	bneck, $3 \times 3$	72	24	-	ReLU	2
$84^2 \times 24$	bneck, $3 \times 3$	88	24	-	ReLU	1
$84^2 \times 24$	bneck, $5 \times 5$	96	40	✓	Hardswish	2
$42^2 \times 40$	bneck, $5 \times 5$	240	40	✓	Hardswish	1
$42^2 \times 40$	bneck, $5 \times 5$	240	40	✓	Hardswish	1
$42^2 \times 40$	bneck, $5 \times 5$	120	48	✓	Hardswish	1
$42^2 \times 48$	bneck, $5 \times 5$	144	48	✓	Hardswish	1
$42^2 \times 48$	bneck, $5 \times 5$	288	96	✓	Hardswish	2
$21^2 \times 96$	bneck, $5 \times 5$	576	96	✓	Hardswish	1
$21^2 \times 96$	bneck, $5 \times 5$	576	96	✓	Hardswish	1
$21^2 \times 96$	conv2d, $1 \times 1$	-	576	-	Hardswish	1
$21^2 \times 576$	avgpool2d, $7 \times 7$	-	576	-	-	1
576	fully-connected	-	256	-	ReLU	-
256	fully-connected	-	1	-	-	-

Таблица 2.1

Структура слоев предлагаемой нейросети

Также на рис.2.3 представлена структура Bottleneck блока. Данный тип элемента можно считать ключевой инновацией, поскольку в нем используются инвертированные остаточные блоки. Особенность таких bottleneck блоков заключается в следующем:

- Инвертированная структура. В традиционных остаточных блоках используется уменьшение размерности данных перед обработкой. В инвертированных блоках сначала происходит увеличение размерности, затем обработка данных, а потом сжатие обратно к исходной размерности. Это позволяет выявлять высокоуровневые признаки в условиях ограниченности вычислительных ресурсов.
- Свертка по глубине. Как уже говорилось в параграфе 2.2.1, depthwise свертка значительно уменьшает вычислительную сложность и количество обучаемых параметров.

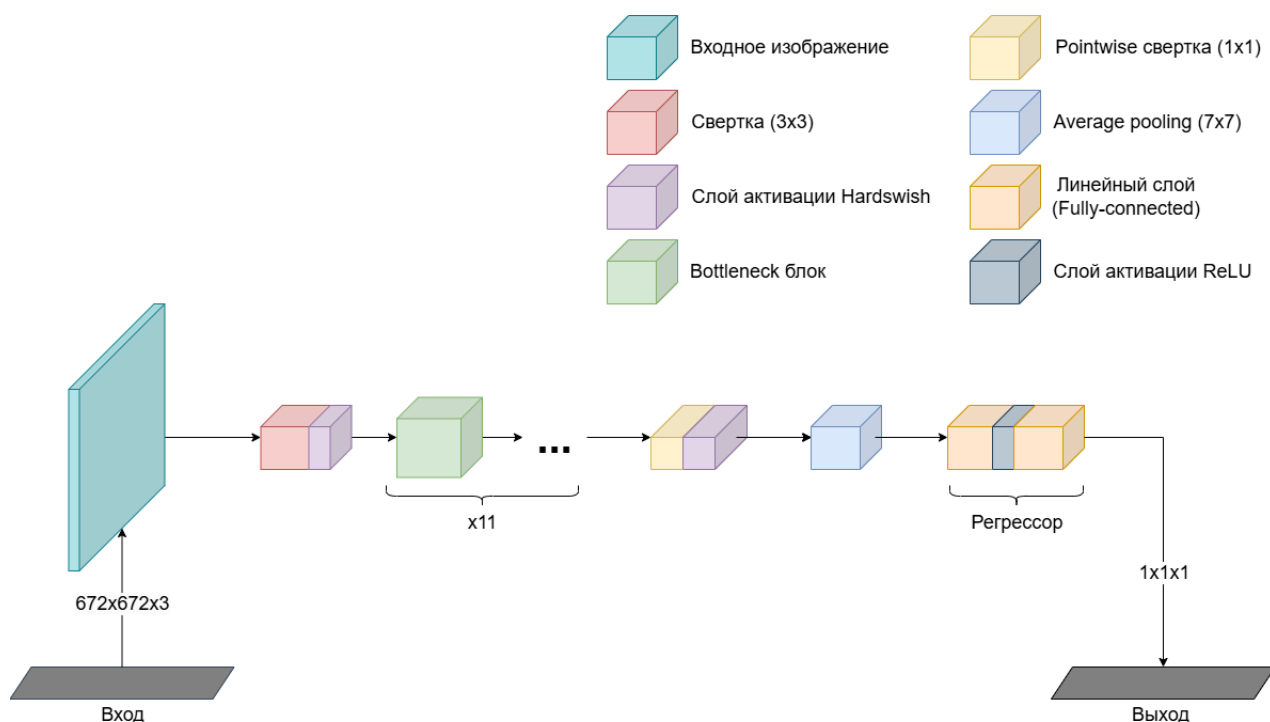


Рис.2.2. Общая схема нейросети

- Squeeze-and-Excitation (SE) блоки. Эти блоки помогают адаптивно перенастраивать весовые коэффициенты каналов, усиливая важные признаки и подавляя менее значимые, что улучшает представление данных и общую производительность модели.
- Функция активации Hardswish. Эта функция проста в вычислительном смысле, помогает нейросети справляться с лениейными зависимостями, хорошо сохраняет информацию и градиенты при отрицательных значениях входных данных.

## 2.4. Определение направления смещения

Попытка смоделировать размытое изображение (например, для обучения нейросети) с помощью простых инструментов таких, как размытие по Гауссу, приводит к тому, что между изображениями по обе стороны от фокальной плоскости, но на равном удалении нет разницы. Соответственно, в таком случае нельзя определить направление смещения камеры от объекта съемки. Однако реальная дефокусировка устроена сложнее, она содержит асимметричные хроматические и сферические aberrации. Изображение, получаемое с микроскопа в силу неидеальности систем регистрации и оптики, а также физики света является сверткой «чистого» изображения и функции рассеяния точки (ФРТ) [7]. Помимо этого

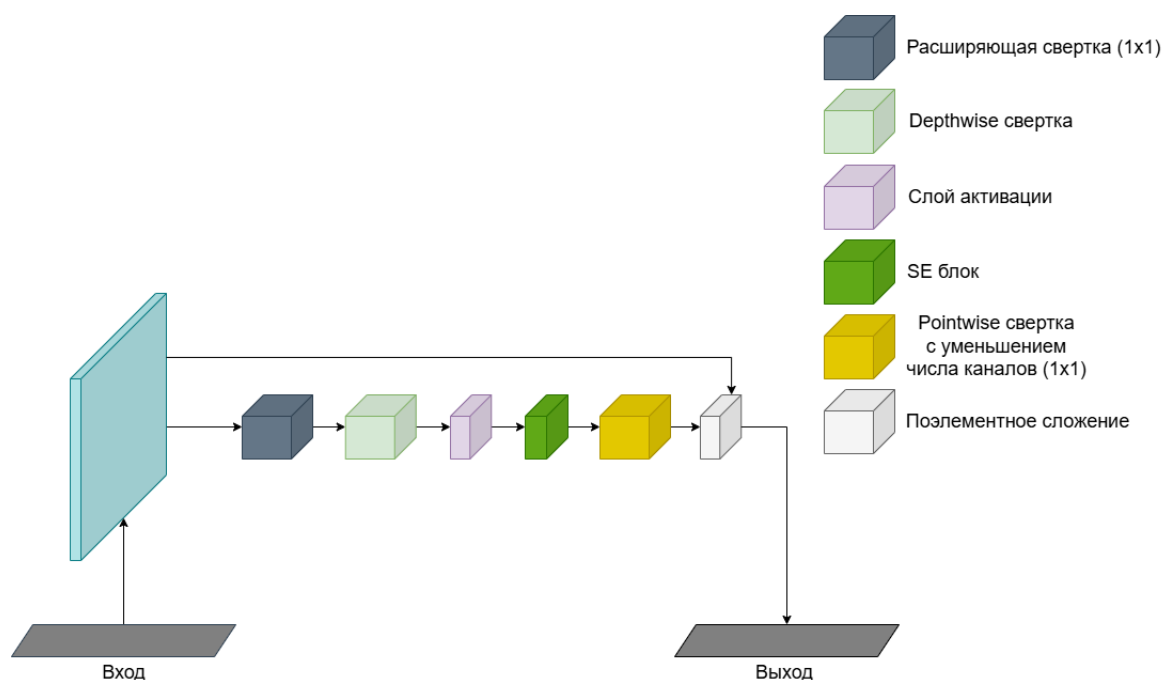


Рис.2.3. Структура bottleneck блока

камера вносит свой шум в получаемое изображение. Таким образом, итоговое изображение можно описать формулой:

$$I' = I \otimes H + N, \quad (2.6)$$

где  $I$  – исходное изображение,  $H$  – ФРТ,  $N$  – шум от системы регистрации,  $I'$  – результирующее изображение.

Благодаря такому механизму, реальные снимки объекта по разные стороны фокальной плоскости будут отличаться. На рис.2.4 приведена оценка асимметрии функции рассеяния точки, смоделированной на разных длинах волн и на разном расстоянии от фокальной плоскости. Хорошо видно, что противоположные друг другу изображения неодинаковые. Именно это позволяет определять направление движения камеры.

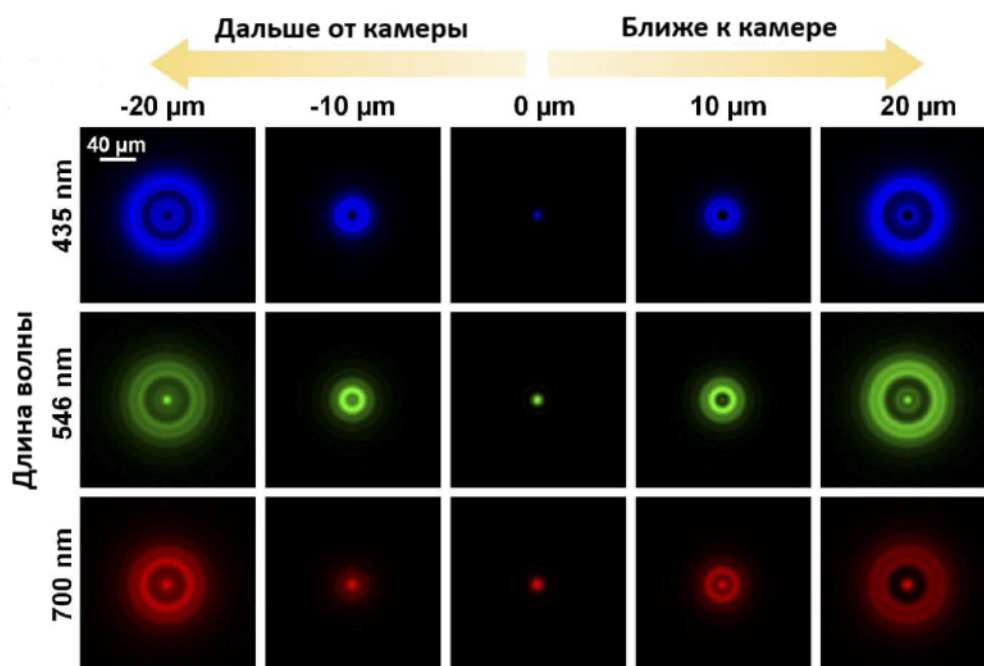


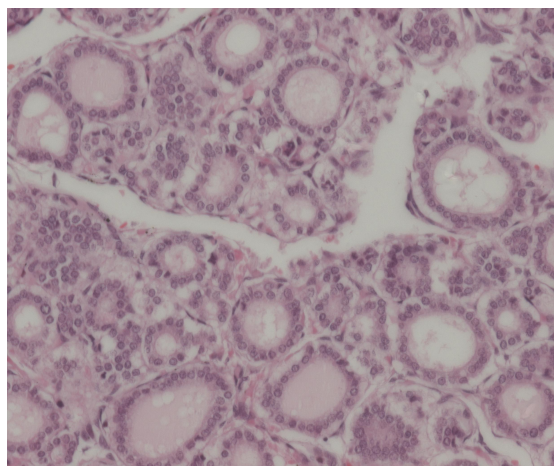
Рис.2.4. Смоделированные ФРТ

## ГЛАВА 3. ОБУЧЕНИЕ НЕЙРОСЕТИ

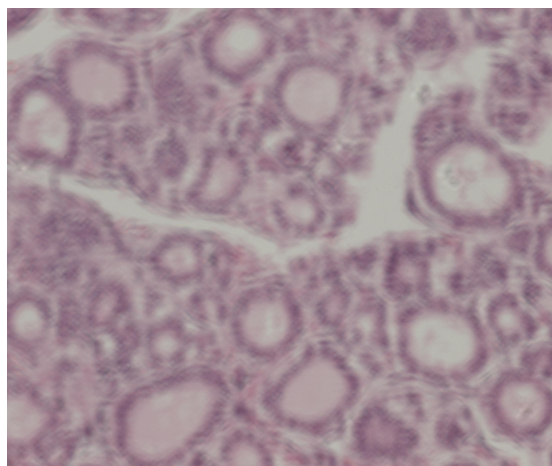
В данной главе будет представлено описание данных с примерами, а также будет рассказано о процессе обучения предлагаемой нейросети и его результатах.

### 3.1. Данные для обучения

Для обучения и тестирования нейросети использовался набор данных (датасет) из открытых источников [9]. В нем представлено более 900 тыс. снимков различных типов объектов с микроскопа разного разрешения и с разным числом каналов. Для обучения нейросети будут использоваться 6 тыс. изображений клеток, отснятых на 20-кратном увеличении. Глубина резкости используемого объектива равна 1 мкм. Данные представлены в виде набора файлов (изображений в формате jpeg) с именами «defocus[N]», где N – число со знаком, означающее отклик, который нужно предсказать. Например, имя файла «defocus-550.jpg» означает, что текущую плоскость, на которой фокусируется камера, необходимо сместить на -550 нанометров. Число -550 – это то значение, которое должно получиться в качестве результата после работы нейросети. Пример входных данных представлен на рис.3.1



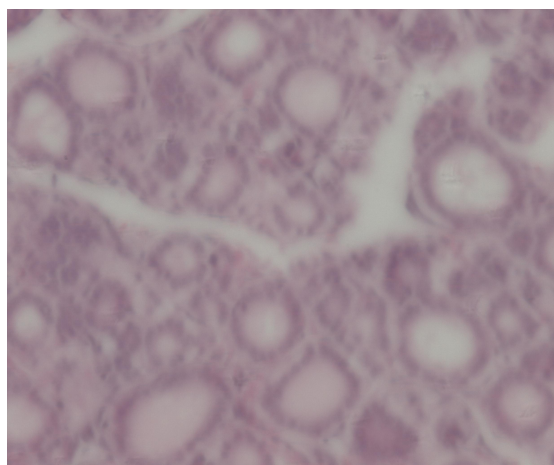
a)



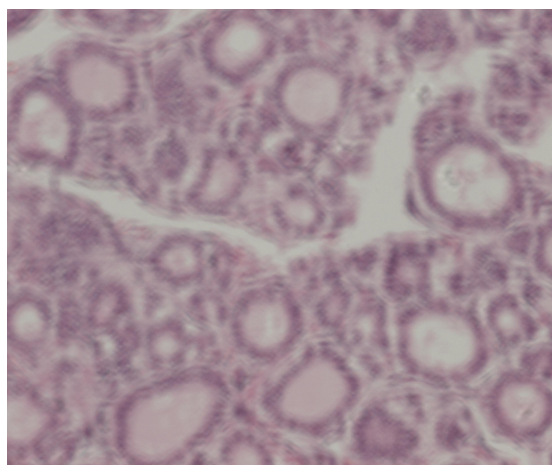
b)

Рис.3.1. Примеры изображений из набора данных: *a* — Расстояние дефокусировки -550 нм; *b* — Расстояние дефокусировки 9550 нм

Реальные отснятые данные также подтверждают тезис о том, что из-за асимметричности ФРТ изображения на равном расстоянии но по разные стороны от фокальной плоскости объекта будут различаться (рис.3.2)



a)



b)

Рис.3.2. Примеры асимметрии ФРТ: *a* — Расстояние дефокусировки -9550 нм; *b* — Расстояние дефокусировки 9550 нм

### 3.2. Результаты обучения

Набор данных был разделен на тренировочную, валидационную и тестовую выборки в соотношении 80:10:10.

*Тренировочная выборка* — часть набора данных, используемая для обучения модели, то есть для настройки весов и других обучаемых параметров ее слоев.



*Валидационная выборка* – данные, которые также используются в процессе обучения, но не влияют на обучаемые параметры. Валидационная выборка используется для корректировки гиперпараметров модели.

*Тестовая выборка* – данные, которые первый раз подаются на вход нейронной сети только после полного процесса обучения. Она используется для оценки качества модели на новых, незнакомых ей данных.

Обучение (или тренировка) производилось на графическом процессоре NVIDIA GeForce RTX 2080Ti с 11 Гб видеопамяти. Для ускорения обучения использовалась подача изображений пакетами (batch) по 30 штук. Тренировка проходила в течение 150 эпох, скорость обучения (learning rate) была динамической с начальным значением 0.001 и уменьшалась в 10 раз, если в течение 5 эпох не происходило улучшение показателей ошибки. Время обучения составляло 1 час. График оптимизируемой функции представлен на рис.3.3.

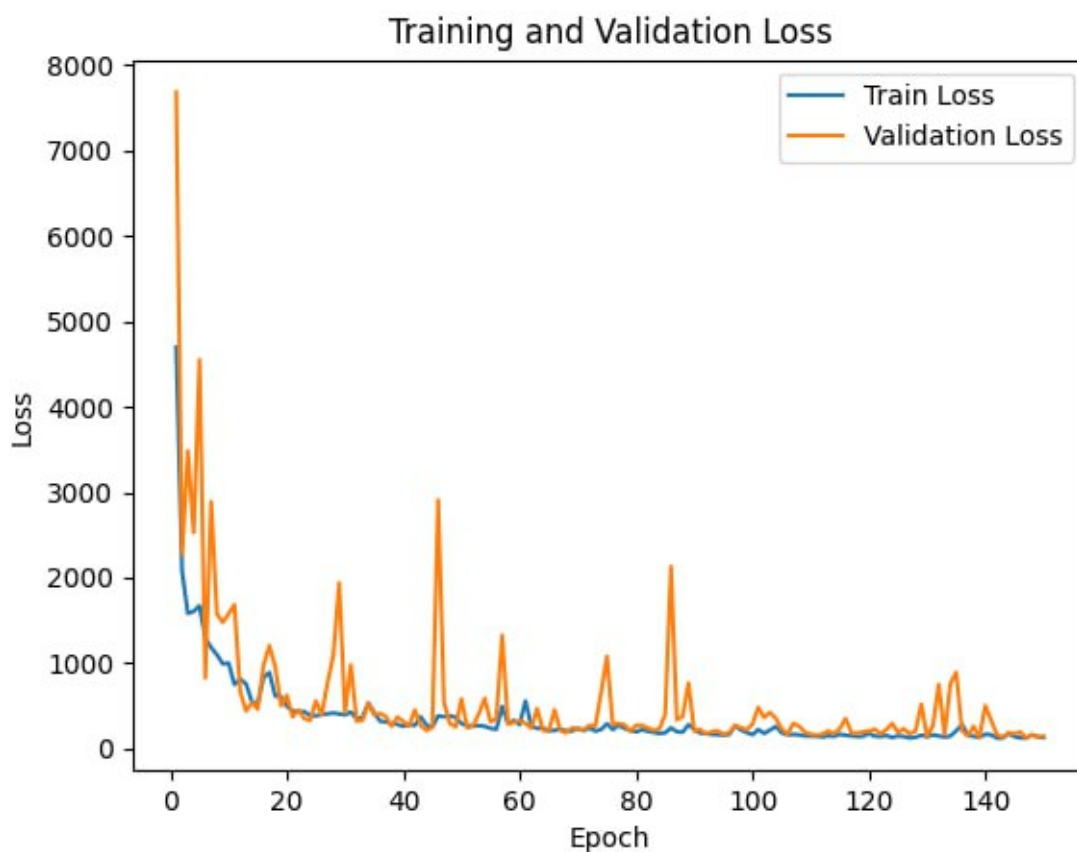


Рис.3.3. Результат обучения. Ось ОХ – номер эпохи, ось ОУ – величина функции потерь

### 3.3. Метрики

Для обучения нейронной сети и оценки ее качества использовалась функция потерь smooth L1-loss, как было сказано в параграфе 2.3. В процессе обучения она минимизируется. Также замеряется время работы алгоритма.

Для оценки результата на тестовом наборе также используется метрика  $MAE$ :

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - y'|, \quad (3.1)$$

где  $y$  – истинное значение отклика,  $y'$  – предсказанное значение,  $N$  – мощность тестовой выборки.

### 3.4. Эксперименты

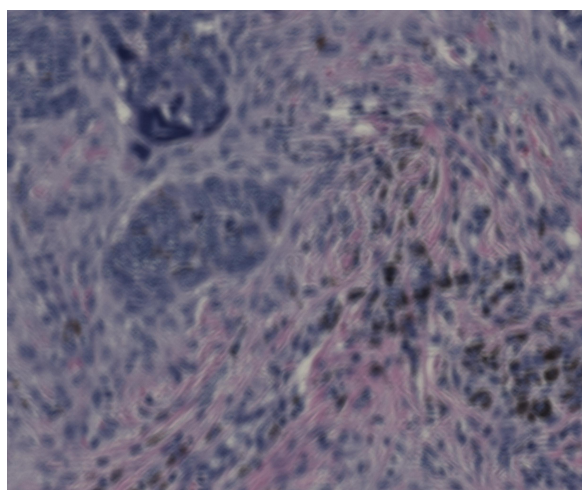
На рис.3.4 представлен результат работы предложенной нейросети при том же увеличении, что и данные для обучения. Хорошо видно, что изображение стало четче, границы объектов различимые и резкие.

Так как одна из целей работы – разработка решения, не требующего применения дополнительных датчиков, то прямое сравнение предлагаемой нейронной сети будет проводиться с контрастным автофокусом и методом анализа резкости. Для этого применим эти алгоритмы к одним и тем же данным.

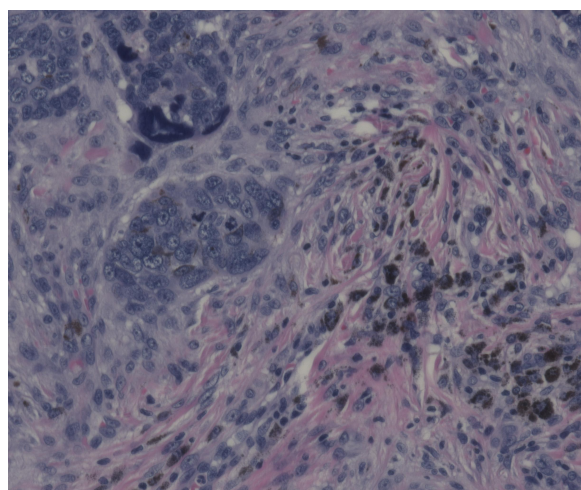
В табл.3.1 представлены средние значения времени работы, ошибки и числа перемещений камеры, необходимого для завершения процесса, для каждого метода. Тестирование производилось на процессоре Intel Core i7-4770 3.4 GHz и видеокарте NVIDIA GeForce RTX 2080Ti. Предлагаемая нейронная сеть лучше справляется с поставленной задачей по всем критериям качества. Кроме того, средняя ошибка оказывается меньше, чем глубина резкости применяемого объектива (1 мкм), а значит, визуально объект будет находиться в фокусе.

Решение на основе глубокого обучения также оказывается более устойчивым к шуму. Это свойство достигается за счет использование сверточных слоев.





a)



b)

Рис.3.4. Примеры асимметрии ФРТ: *a* — до фокусировки; *b* — после фокусировки

Метод	Время расчета одного смещения, мс	Ошибка, мкм	Число перемещений камеры	Общее время, мс
Контрастный автофокус	29.0	6.3	13.4	1058.6
Метод анализа резкости	27.5	5.2	11.2	871.4
Нейросеть (CPU)	21.1	<b>0.9</b>	<b>1</b>	71.1
Нейросеть (GPU)	<b>7.2</b>			57.2

Таблица 3.1

Сравнение классических методов и предлагаемой нейронной сети

## ЗАКЛЮЧЕНИЕ

В данной работе были изучены существующие классические и нейросетевые методы автоматической фокусировки цифровых микроскопов. В ходе исследования было заключено, что нейросетевые подходы в этой области только начинают развиваться и широко используемых решений еще нет. Исходя из этого, была предложена нейронная сеть на основе глубокого обучения, решающая поставленную задачу. Резюмируя изложенный в главах 2 и 3 материал, можно заключить следующее:

- Предложенный метод нейросетевого автофокуса за один шаг позволяет добиться лучшей точности, чем классические методы. С методами, использующими фазовые датчики, нейросеть можно сравнить лишь косвенно,

однако, опираясь на материалы статьи, можно также сделать вывод, что фазовые методы уступают предложенному решению.

- Предложенное нейросетевое решение имеет более высокую скорость работы. Во многом это достигается именно благодаря отсутствию промежуточных перемещений камеры.
- Алгоритм устойчив к шуму, возникающему во время съемки, что является еще одним его преимуществом.
- Предлагаемое решение можно развернуть на большинстве комплексов, так как данный метод не требует дополнительного оборудования, а большинство лабораторий применяет моторизованные микроскопы совместно с персональными компьютерами.

Стоит отметить, что для применения алгоритма в других областях может потребоваться дополнительное дообучение нейросети, поскольку нейронные сети при обучении выучивают основные признаки обучающего набора данных. Однако способность определять расстояние до фокальной плоскости остается, но точность в таком случае падает.

## СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

- ФРТ**    Функция Рассеяния Точки.
- PDAF**    Phase Detection Autofocus.
- MSE**    Mean Squared Error.
- MAE**    Mean Absolute Error.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Dastidar T. R., Ethirajan R.* Whole slide imaging system using deep learning-based automated focusing // *Biomed. Opt. Express.* — 2020. — Т. 11, № 1. — С. 480—491. — DOI 10.1364/BOE.379780. — URL: <https://opg.optica.org/boe/abstract.cfm?URI=boe-11-1-480>.
2. *Goodfellow I., Bengio Y., Courville A.* Deep learning. — MIT press, 2016.
3. *Ho C.-J., Chan C.-C., Chen H. H.* AF-Net: A Convolutional Neural Network Approach to Phase Detection Autofocus // *IEEE Transactions on Image Processing.* — 2020. — Т. 29. — С. 6386—6395. — DOI 10.1109/TIP.2019.2947349.
4. *Hu J., Shen L., Sun G.* Squeeze-and-excitation networks // *Proceedings of the IEEE conference on computer vision and pattern recognition.* — 2018. — С. 7132—7141.
5. Mobilenets: Efficient convolutional neural networks for mobile vision applications / A. G. Howard [и др.] // *arXiv preprint arXiv:1704.04861.* — 2017.
6. Rapid whole slide imaging via learning-based two-shot virtual autofocusing / Q. Li [и др.] // *arXiv preprint arXiv:2003.06630.* — 2020.
7. *Sibarita J.-B.* Deconvolution microscopy // *Microscopy Techniques: -/-.* — 2005. — С. 201—243.
8. *Sobel I., Feldman G. M.* An Isotropic 3×3 image gradient operator //. — 1990. — URL: <https://api.semanticscholar.org/CorpusID:59909525>.
9. Transform-and multi-domain deep learning for single-frame rapid autofocusing in whole slide imaging / S. Jiang [и др.] // *Biomedical optics express.* — 2018. — Т. 9, № 4. — С. 1601—1612.
10. *Павельева Е. А.* Обработка и анализ изображений на основе использования информации о фазе // *Компьютерная оптика.* — 2018. — Т. 42, № 6. — С. 1022—1034.