将股票价格走势预测视为图像分类问题

Matej Steinbacher

arXiv:2303.01111v1 [q-fin.PR] 2 Mar 2023

摘要——**本文将股票的日内价格走势视为图像分类问题进行研究。我们使用基于卷积神经网络的模型，有力地论证了交易的第一个小时与收盘价之间存在高层次关联。该算法成功地区分了两个相互对立的类别；根据算法预测进行投资，表现优于所有其他方案，仅次于理论最大值。为支持这一论点，我们还开展了若干额外测试。本文的研究结果凸显了计算机视觉技术在金融市场研究中的适用性，尤其是在预测股票价格走势方面。**

索引词——**深度卷积神经网络、MobileNet-V2、图像分类、股票价格、投资问题。**

I. 引言

GAINING 对于学术界和专业人士而言，能够预测股票及其他可交易资产的价格走势仍至关重要。市场波动剧烈，且本质上受各种经济、政治和心理因素影响，因此这类预测颇具挑战，尤其是在日内层面。例如，2022年9月21日，在备受瞩目的美联储主席杰 Jerome Powell有关加息的讲话中，纳斯达克指数最初在前30分钟内下跌近1.5%，随后30分钟内上涨超过2%；再接下来的30分钟内又回落近1.7%，最终当天收于低点，额外下跌1.3% [1]。

为了解决这一问题，人们提出了许多不同的金融和经济变量作为潜在预测指标，例如股息价格比、股息率、市盈率、账面市值比等估值比率，各种利率和利差，以及包括通货膨胀和工业生产在内的宏观经济变量。参见例如[2]–[9]。 [10] 对心理学在资产定价中的作用作了很好的总结。这些论文帮助我们从根本上理解金融市场，至多解释了变量之间的一般关系，但事实证明，它们在预测个股价格走势方面用处有限。

近来，人工智能领域，尤其是计算机视觉领域的总体进展也开始延伸至金融市场领域。例如，有研究 [11]将一维卷积神经网络模型与双向LSTM单元结合起来构建流水线。 [12]在限价订单簿（即买入和卖出订单列表）上使用卷积神经网络模型，以预测价格变动的方向。 [13] 使用 LeNet-5 模型，基于标准普尔500指数数据预测股价变动的布尔方向。作者使用分钟级收盘价，并计算了九种技术指标作为输入变量，再将这些指标 30 分钟的数据绘制成图像。类似地， [14]计算了 28 天内的 28 种技术指标，生成一个 28×28 方阵，并使用基于卷积神经网络的模型进行处理；或者， [15] 使用了 15×15 帧。 [16]将卷积神经网络与循环神经网络神经元的元素整合进一种多滤波器结构，以分析中国股票市场指数沪深300； [17]则提出了一种图卷积神经网络，将总体市场信息与个股信息结合为联合特征。关于更全面的综述，请参见 [18], [19]。

本文基于纳斯达克市值最大的股票的数据，研究日内股票收益率的可预测性。我们将股票交易决策视为图像分类问题，将所有日内收益划分为三个类别，分别表示买入、卖出或波动过大、无法判断。该模型基于 MobileNet-V2 [20] 构建。为向模型提供输入，我们将所有金融数据可视化为图像。我们对问题的构建基于这样一个假设：交易的第一个小时与收盘价之间存在稳定的高层次关系。

图像分类是计算机视觉中的一个标准领域。其方法基于这样一种假设：处理大量图像可以从数据中提取某种高层抽象，加深对相关问题的理解，并（希望）促进决策。本文探讨的基于卷积神经网络的模型，相较于传统神经网络有三大优势：参数共享、稀疏交互和等价表示 [21]。为了充分利用输入数据的二维结构，网络采用局部连接和共享权重，而非传统的全连接网络。

仿真结果有力地证明了我们的方法具有优势：按照算法预测进行投资，表现优于所有替代架构，仅次于算法的理论最大值。在包含三个类别的样本中，测试的总体准确率分数为 51.5%；针对规模进行校正后，两个最重要类别 C1 和 C2 的精确率分数分别为 53.3% 和 58.3%。算法很好地区分了处于对立两端的类别，其绝大多数错误分类都被归入第三个中性类别 C0；该类别作为两者之间的“缓冲区”而加入。进一步收紧分类条件提高了

M. Steinbacher 是一名独立研究人员，隶属于 Pixlifai。联系邮箱: matej.steinbacher@gmail.com。

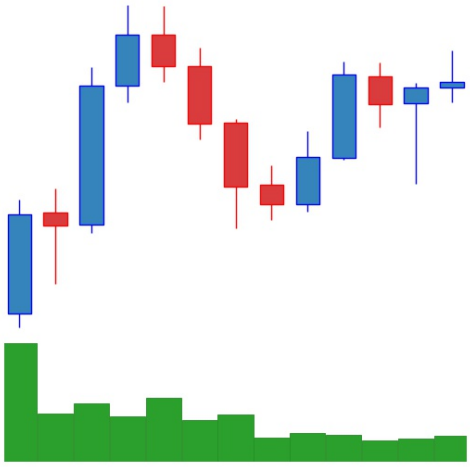

图1. 样本图像

分数。例如，要求“认可”分数至少达到 95% 才作出判断，可使准确率达到 88.7%。然而，它只对 71 个观测值进行了分类，占原始数据集的 2.2%。这清楚地表明，这是以数量换取质量。总而言之，结果有力地证明，按照论文中的方法，使用基于深度图像的模型对金融市场建模是可行的。结果还表明，有必要重新思考在深度神经网络建模中传统使用的性能度量，因为这些度量似乎并不充分。

本文结构如下。第2章介绍数据。模型使用纳斯达克上市股票中市值最大的股票数据构建。第3章概述模型。我们采用 Google 的 MobileNet-V2 作为基础模型。第4章进行了大量测试和模拟。第5章展示并讨论实时测试的结果。最后一章总结全文。

II. 数据

初始数据集由按市值衡量的纳斯达克上市公司中规模最大的1000家组成。数据收集时间为1月14日至2022年7月1日，使用 Python 的开源库 yfinance [22] 收集，并使用 Python 的开源库 mplfinance [23]进行可视化。股票常规交易时间为东部时间周一至周五上午9:30至下午4:00。数据按5分钟的时间间隔收集，每个时间间隔均采用开盘价、收盘价、最低价、最高价和成交量信息。所有不完整的数据单元均已删除。

每个数据单元 $s_i = \{x_i, y_i\}$ 由一对输入图像 x_i 和对应标签 y_i构成。输入图像样本如图 1 所示。

图中展示了常规交易时段第一个小时内每个5分钟间隔对应的一系列蜡烛图及成交量柱。蜡烛图是一种用于技术分析的价格图表，显示资产在指定时间间隔内的最高价、最低价、开盘价和收盘价。交易者常借助蜡烛图寻找价格形态。成交量以柱状图的形式绘制，显示资产在指定时间段内的交易股票份额。

数据单元按照以下关系进行标记：

$$y_i = \begin{cases} 1, & \text{if } \frac{p_T^i}{p_0^i} \geq 1.02, \\ 2, & \text{if } \frac{p_T^i}{p_0^i} \leq 0.98, \\ 0, & \text{otherwise.} \end{cases}$$

其中，y_i 是观测值 i的标签，p_0^i 表示资产 i 在常规交易开始后一小时的收盘价，而 p_T^i 是资产 i在常规交易时段收盘时的收盘价。

数据集被划分为三个互不重叠的子样本，随后分别用于训练、验证和测试。为控制类别不平衡，通过随机删除数据单元对各子样本进行下采样。表 I 展示了各子样本的汇总统计量。

III. 模型

我们使用了 MobileNet-V2（MNET-2） [20]。这是一种小型卷积神经网络模型，结合了深度可分离卷积和倒置残差块，其中捷径连接位于较窄的瓶颈层之间。整个架构使张量的尺寸保持相对较小，尽管通道数有所增加，但空间维度很小；这主要得益于瓶颈层，它们使块之间的连接保持在低维空间中。该模型基于这样一个前提：“感兴趣流形”可以嵌入低维子空间。选择该模型主要有三个原因：作为参数量较少的轻量级模型，它有望降低过拟合的可能性；此外，它的计算成本较低且性能良好。

模型在原始模型最后一个线性层之上进行训练，采用 20% 的丢弃正则化率。最后的全连接层没有非线性，并直接连接到用于分类的 Softmax 层。Softmax 函数将 logit 向量映射为后验概率分布。模型使用随机梯度下降（SGD）进行优化，学习率为 0.001，每经过 30 个训练轮次便衰减为原来的十分之一，动量为 0.9，权重衰减为 0.0001，并采用 Nesterov 动量。

输入图像具有固定的分辨率 224×224 和 3 个颜色通道，这是模型所要求的；每个像素均定义在 $[0, 255]$ 域上。数据集已使用 $x' = (x/255)_{\sigma}^{-\mu}$ 进行归一化，其中 $\mu = [0.485, 0.456, 0.406]$ 且 $\sigma = [0.229, 0.224, 0.225]$。

整个建模过程均在 PyTorch 中完成，并在一块配备 2GB 内存的 Nvidia GeForce 930M 图形处理器上运行。每个模型训练了 $T = 50$ 个训练轮次。结果如图2所示。

图2展示了训练阶段（实线）和验证阶段（虚线）的准确率。表现最佳的模型在验证集上的准确率达到 49.36%。鉴于输入数据分为三个平衡类别，该结果整体上比随机猜测高出 48%。

两条曲线的形状都表明，训练存在过拟合现象。过拟合模型可能无法恰当地

表 I 数据集汇总统计量

	S_l				S_v				S_t			
	ALL	0	1	2	ALL	0	1	2	ALL	0	1	2
AVG	1.000	1.000	1.033	0.968	1.000	1.000	1.033	0.968	1.003	1.000	1.033	0.969
中位数	0.999	1.000	1.028	0.972	1.000	1.000	1.028	0.972	1.002	1.000	1.028	0.973
SD	0.029	0.010	0.014	0.013	0.029	0.009	0.014	0.012	0.020	0.010	0.015	0.013
MIN	0.745	0.980	1.020	0.745	0.875	0.980	1.020	0.875	0.876	0.980	1.020	0.876
MAX	1.205	1.020	1.205	0.980	1.127	1.020	1.127	0.980	1.152	1.020	1.152	0.980
Q1	0.977	0.993	1.024	0.964	0.977	0.993	1.024	0.964	0.992	0.993	1.024	0.966
Q3	1.023	1.007	1.036	0.977	1.023	1.007	1.037	0.977	1.013	1.007	1.038	0.977
N	14,175	5,000	4,457	4,718	3,983	1,400	1,255	1,328	3,179	2,314	566	299

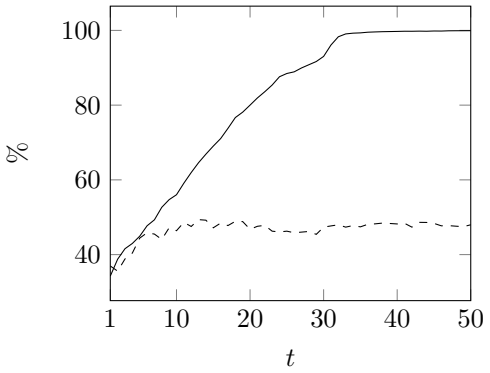

图2. 训练结果

表 II 混淆矩阵

真实值 / 预测值	0	1	2	SUM
0	1200	728	386	2314
1	185	324	57	566
2	131	56	112	299
SUM	1516	1108	555	3179

表 III 性能度量

	精确率	召回率	F1 分数	支持度
0	0.79	0.52	0.63	2314
1	0.29	0.57	0.39	566
2	0.20	0.37	0.26	299

泛化其本应学习的特征，反而会拟合训练样本本身的特异性。此类模型在训练期间表现良好，但对于除其训练所用数据之外的任何其他数据，表现都不尽如人意。过拟合是监督式机器学习领域的一个普遍问题，且无法避免 [24]。

为抑制过拟合的影响，我们在分类前的网络末端添加了一个随机失活层，随机丢弃 20% 的单元。随机失活通常被认为是对抗过拟合的非常有效的方法 [25], [26]。

IV. 测试与仿真

A. 测试

测试结果如表 II 和表 III 所示。该算法成功正确分类了 3179 个观测值中的 1636 个，总体准确率为 51.46%。尽管最大类别 C0 占观测值的 72.8%，该算法却将其低估，仅将全部预测结果中的 47.7% 分为该类别。另一方面，C1 相对于另外两个类别的重要性有所提升。该算法较好地区分了相反类别 C1 和 C2，并将大多数错误分类归入中性类别 C0。例如，C2 中只有 18.7% 被误分为 C1，而 43.8% 被误分为 C0。此外，C1 的观测值中只有 10% 被分为 C2，而 32.7% 被分为 C0。

尽管总体准确率水平是最直观的性能指标，但对我们而言，它的相关性较低，主要有两个原因：（1）数据集不平衡；（2）我们对不同类别及其预测的关注程度不同。首先，如果所有观测值都被归为 C0，算法就能获得 72.8% 的准确率。其次，将 C0 错误预测为 C1 虽然不理想，但影响较小，因为 C0 的分布相当对称，预期收益为 0%。另一方面，将 C2 错误预测为 C1 则会导致不可避免的损失，至少为 2.5%，平均为 3.1%。

为了计算平均收益，我们将表 II 中的结果与表 I 中的数据结合起来。其值为 $E(\gamma|\pi,\sum_{k=1}^{J}\pi_{k\,1}\gamma_k = 0.8 \qquad \gamma = [\gamma_1 \ \dots \ \gamma_J]$ 于%，类别 k 和 $\pi_{i,1}\overline{\ } = [0.657\ 0.292\ 0.05\overline{1}]$的平均收益分别为。该值高于数据集的平均值 0.3%。

B. 宏观分析

假设每个类别 C_j 均服从 F_j，并令 $\pi_{i,j}$ 表示算法将类别 i 的产量判定为类别 j 的概率，其中 $\pi_{i,j} \in [0,1]$ 且 $\sum_{j=1}^{J}\pi_{i,j} = 1 \forall i$。进一步假设每个类别 j 都有各自的投资倾向 $\alpha_j \in [0,1]$。那么，从 X 中随机抽样所得的预期收益可表示为

$$E(\gamma_i|x_i \in X) = \sum_{j=1}^{J}\left(\frac{|C_j|}{\sum_{k=1}^{J}|C_k|}E(X|F_j)\cdot\pi_{j,k}\cdot\alpha_k\right),$$

其中，$k \in \{1,...,J\}$ 和 $|C_j|$ 是类别 j的规模。我们可以证明，存在某种类别规模组合，使预期收益为正。例如，假设有 3 个类别，其预期收益如下：$\gamma_1 \geq 0$ 和 $\gamma_3 \leq 0$，而 γ_2 可以取任意值。$n_i = \frac{|N_i|}{\sum_{k=1}^{3}|N_k|}$ 是类别i 观测值的比例，并且只让被分类为 C_1 的样本进行投资，或者令 $\alpha_1 = 1$、$\alpha_2 = 0$ 和 $\alpha_3 = 0$ 表示投资倾向，则可写为

$$n_1 \cdot \gamma_1 \cdot \pi_1 + n_2 \cdot \gamma_2 \cdot \pi_2 + n_3 \cdot \gamma_3 \cdot \pi_3 \geq 0,$$

满足 $n_1 + n_2 + n_3 = 1$的约束。 π_i 表示类别 i 被分类为类别 1 的概率。

然后， $n_1 \geq -\frac{\gamma_2 \cdot \pi_2}{\gamma_1 \cdot \pi_1} \cdot n_2 - \frac{\gamma_3 \cdot \pi_3}{\gamma_1 \cdot \pi_1} \cdot n_3$ 可以改写为

$$\frac{n_1}{n_2} \geq -k_2 - k_3 \cdot \frac{n_3}{n_2}.$$

k_2 通常会非常小，接近 0，而 k_3 ≤0。这意味着，正收益类别的规模n_1 应与负收益类别 n_3的规模成比例。例如，如果正收益的预期收益 $\gamma_1 = 0.03$ 而负收益的预期收益 $\gamma_3 = -0.03$，那么在其他条件相同的情况下， n_1 至少需要约为 n_3 的三分之一，才能在平均意义上达到收支平衡。

为了检验上述推导，我们进行了三批蒙特卡罗模拟，每批重复 10,000 次。 X_j 类别 j =1，.....， J服从截断正态分布 X_j ~$N(\mu_j,\ \sigma_j^2;\ a_j,\ b_j)$，其均值为 μ_j，方差为 σ_j^2，且 $-\infty \leq a_j < b_j \leq \infty$。模拟中使用的参数列于表 IV。

表 IV 蒙特卡罗参数

类别	μ	σ	a	b	Exp1	Exp2	Exp3
1	0.03	0.015	0.02	0.15	33	100	50
2	0.0	0.01	-0.02	0.02	10	10	10
3	-0.03	0.015	-0.15	-0.02	100	100	300

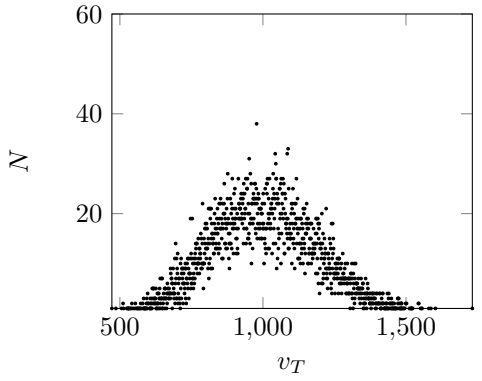

图3. 蒙特卡罗模拟：实验1

所有情况下的初始投资均为1000美元，后续投资均等于 min(v_t，1000)，具体金额取决于相应的概率 $\pi_{i,1} = [0.572, 0.315, 0.187]$ ，这些概率根据表 II 计算得出。 v_t 是截至时间 t的累计总和。仿真结果见图3、图4、图5。

图3展示了所有类别的结构均符合闭式数学解时的仿真结果。也就是说，在其他条件不变的情况下，若要达到盈亏平衡，正收益类别1的规模应为负收益类别3规模的三分之一。 $E(x \in C_2) = 0$ 不会影响结果，因此保持不变。运行蒙特卡罗模拟后，初始的1000美元平均经过143次抽样变为1003美元，

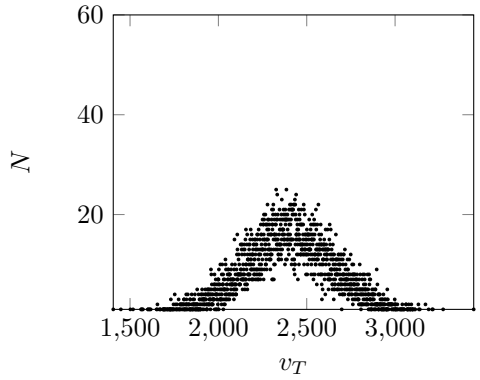

图4. 蒙特卡罗模拟：实验2

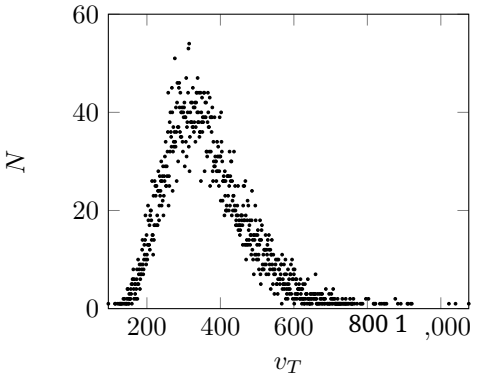

图 5. 蒙特卡罗模拟：实验3

每次抽样的收益率微小，为 $E(x \in X|$实验1) $= 0.002\%$。该结果基本符合理论解。

图 4 展示了在其他条件不变的情况下，正收益和负收益类别数量相同的仿真结果。平均而言，这次运行将初始投资转化为 2400 美元。其结果显然是两类买入概率不同所致，而且买入概率明显偏向真实买入类别。最后，在图 5 所示的看跌日度情绪下，这次运行造成了重大亏损，平均而言，初始的 1000 美元降至 361 美元。由此可见，在其他条件不变时，也存在这样一种状态：使用算法进行交易平均会导致累计亏损，此时最好完全停止交易。

C. 微观分析

上一节中的所有数值均由相应分布人为生成。本节将使用测试数据集中的全部数据，这些数据反映了真实交易中的情况。

表 V 总结了算法在测试期间的表现。在 3179 个观测值中，1516 个被分类为 C0，每个元素的平均收益为 0.1%；1108 个被分类为 C1，每个元素的平均收益为 0.95%；555 个被分类为 C2，每个元素的平均损失为 -0.45%。

将 C0 的小幅收益视为零收益的一阶近似后，计算出的方向符合预期，这意味着该算法成功地整体区分了不同类别。然而，它大幅

表 V 算法预测：统计数据

	C0	C1	C2
AVG	1.0011	1.0095	0.9955
中位数	1.0000	1.0070	0.9964
SD	0.0171	0.0216	0.0209
MIN	0.9040	0.9330	0.8756
MAX	1.1080	1.1520	1.1217
Q1	0.9920	0.9958	0.9829
Q3	1.0100	1.0230	1.0064
N	1516	1108	555

压缩了幅值。例如，预测的 C1 的平均增益至多为真实 C1 的 28.8%。

只有 C1 预测会触发活动。总共进行了 1108 次 C1 预测，其中 722 次收益非负，386 次收益为负。负收益可能是错误预测为 C0 或 C2 的结果。C1 预测的平均每笔交易收益略低于 1%。这一数值高于刚刚计算出的“宏观”层面平均每笔交易收益（0.8%），也显著高于测试数据集的平均收益（0.3%）（表 I）。

D. 预测分布

预测类别的分布如图 6、7 和 8 所示。频率基于离散化数据计算，增量设为 0.001 或 0.1%。

C0 的形状看起来相当对称，并以 1.0 为中心，偏度为 0.027，但并非来自正态分布。[1] C1 呈现正偏（0.845），大多数数值为正；C2 呈现负偏（-0.024），其分布大多落在负收益率范围内。

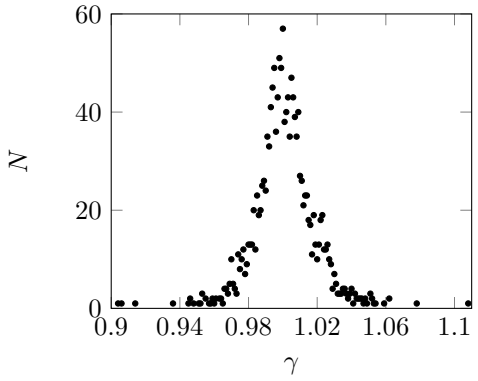

图 6. 类别预测的分布：C0

E. 类别预测与收益率

到目前为止，类别预测概率在特定类别层面被视为外生变量。本节将不再采用这一假设。为了进行测试，我们需要预处理数据。首先，为了限制潜在异常值的影响，将按如下方式截断数据

[1]Shapiro-Wilk $W = 0.966986$, with $p = 4.06126e - 18$.

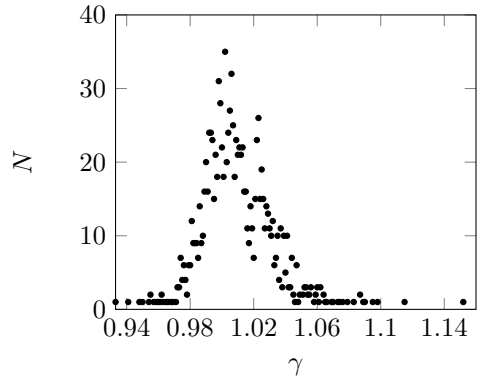

图 7. 类别预测的分布：C1

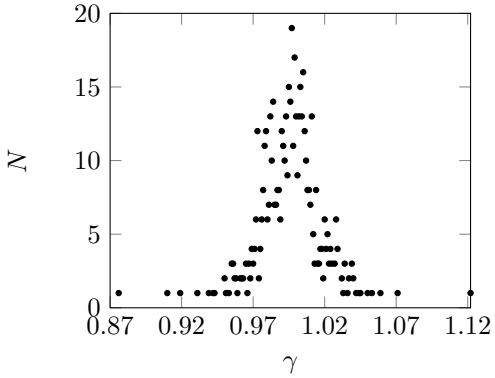

图 8. 类别预测的分布：C2

$$f(x|a = 0.95, b = 1.05) = \begin{cases} a, & \text{if } x \le a, \\ b, & \text{if } x \ge b, \\ x, & \text{otherwise.} \end{cases}$$

其次，为了解决尾部中尤为明显的规模不平衡问题，我们将对数据进行变换，并计算算法预测为各类别的每种产量所占的比例。结果如图 9、图 10 和图 11 所示。

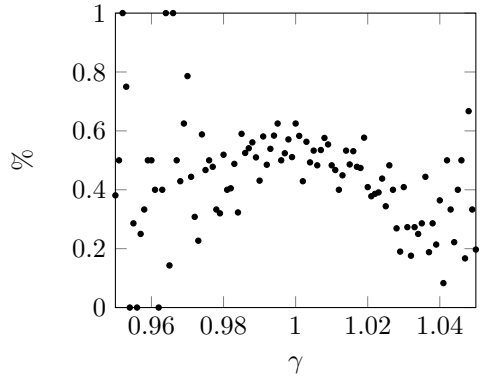

图 9. 按产量划分的类别预测比例：C0

圆点表示每种预测类别在该产量中的比例。例如，C0($\gamma = 0.952$) $= 1.0$ 的值表示

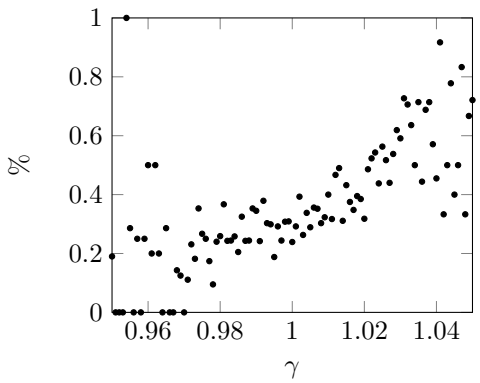

图10. 按产量划分的类别预测比例：C1

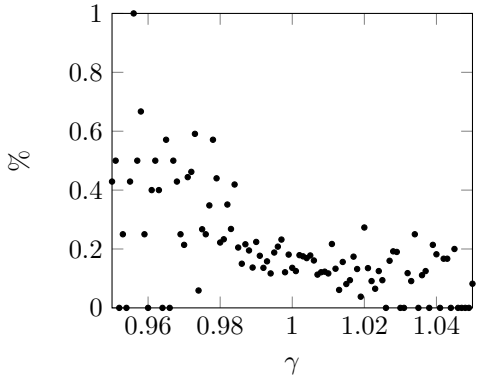

图11. 按产量划分的类别预测比例：C2

该产量对应的所有预测都被归为 C0。[2] 由于数据以相对量表示，因此容易受到异常值的影响，尤其是在某个产量对应的观测值较少时。无论异常值如何，图表都清楚地表明，较高的产量确实更常与 C1 预测相关，而负收益率则更常与 C2 预测相关。

*1) OLS*回归：让我们使用简单线性回归模型，检验每个类别的产量与预测类别比例之间的关系

$$y_{j,i} = \beta_{j,0} + \beta_{j,1} \cdot x_i + \epsilon_{j,i}.$$

变量 x_i 表示观测值 i 的产量，而 $y_{j,\ i}$ 表示被归入某个类别的产量比例$j = (0，1，2)$。模型使用OLS估计量进行估计。括号内为p值。

$$y_0 = \underset{(0.0158)}{1.9251} - \underset{(0.0577)}{1.4878}x \quad \overline{R}^2 = 0.048$$

$$y_1 = \underset{(0.0001)}{-4.6905} + \underset{(0.0001)}{5.0495}x \quad \overline{R}^2 = 0.493$$

$$y_2 = \underset{(0.0001)}{3.7654} - \underset{(0.0001)}{3.5617}x \quad \overline{R}^2 = 0.344$$

估计结果表明，产量与类别预测之间存在线性关系，且具有统计显著性

[2]在这个具体案例中，1.0 的得分仅来自 1 个被（错误地）预测为 C0 的观测值。

对于类别 C1 和 C2，线性模型被证明不适用于预测 C0。这意味着，越是深入分布两端，类别预测越有可能是正确的，即预测为 C1 或 C2。因此，外生给定的类别预测概率无法充分体现不同产量对结果的影响。因此，应采用更复杂的模型分别刻画每个类别的类别预测与产量之间的依赖关系。

2) 多项*Logit*：为补充OLS回归并完成本节，我们运行了一个形式如下的简单多项Logit模型

$$y_{j,i} = \beta_{j,0} + \beta_{j,1} \cdot x_{1,i,j} + \epsilon_{j,i}$$

其中 $y_i \in \{0 = \text{C0}，\ 1 = \text{C1}，\ 2 = \text{C2}\}$ 表示预测类别，x_i表示观测值 $i = (1，\ ...，\ N)$ 和 $j = (0，\ 1，\ 2)$ 的产量。回归结果见表VI。C0 被视为基准选项。

表 V 多项Logit回归摘要

	AVG	预测 = C1	预测 = C2
coef		-22.8763 (0.0000)	14.9750 (0.000)
产量	1.003	22.4499 (0.0000)	-16.0069 (0.000)
LL			-3151.802
对数似然比检验：χ^2			214.596 (0.0000)

结果与普通最小二乘估计一致，并证实产量增加与有利于预测 C1（相对于 C0）的优势比显著提高，以及预测 C2（相对于基准类别 C0）的优势比显著降低相关。这意味着，产量增加会提高预测为 C1 的概率，而产量减少会提高预测为 C2 的概率。为了更清楚地说明这一点，我们在更宽的范围内计算了全部三个类别的概率预测，并在图12中展示结果。

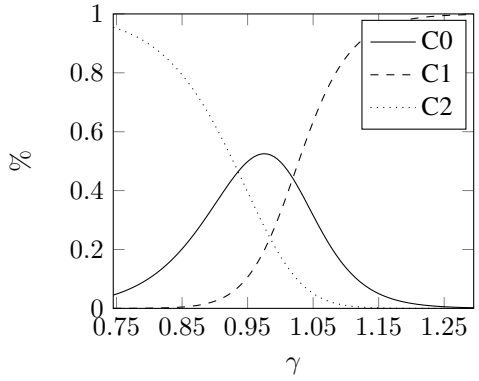

图12。类别概率预测

F. 类别级分析

让我们检查一下，目前的研究结果是否也适用于特定类型预测的层面。每个产量 x_i 都使用因子 m 进行舍入，以得到 $x_i = m\lceil \frac{x_i}{m} \rfloor$。 m 的设定是在舍入造成的舍入误差与创建有意义的分析分组之间进行权衡。我们

认为 $m = 0.001$ 或 0.1% 是两者之间的良好折中。

令 $T(x_{i,j}^T \in X)$ 和 $P(x_{i,j}^P \in X)$ 分别为观测产量和预测产量的两个多重集 x，其中 $i = (1, \ldots, N)$ 且 $j \in \{0, 1, 2\}$。$x_{i,j}^T = x_{i,j}^P$ 表示一次正确预测。

令 $C_j^T(\gamma)$ 表示所有类别为 T_j 且产量等于 γ 的观测值数量，对所有类别 $j = (0, 1, 2)$ 均如此；令 $C_j^P(\gamma)$ 表示所有类别为 P_j 且产量等于 γ 的预测数量，对所有类别 $j = (0, 1, 2)$ 均如此。则 $y_{j,k}(\gamma) = \frac{\gamma}{T_j}$，其中 $k = (0, 1, 2)$ 和 $j = (0, 1, 2)$ 表示每种产量 γ 下类别 j 的 k 类别预测比例。

让我们用以下形式的简单线性回归模型来检验这些关系：

$$y_i = \beta_0 + \beta_1 \cdot x_i + \epsilon_i$$

其中，变量 x_i 表示产量，y_i 表示相应的 $y_{j,k}(\gamma)$。使用 OLS估计量估计方程，结果见表 VII，括号内为 p 值。[3]

回归系数的符号和幅值与先前的预期一致。也就是说，产量上升与观测值被归类为 C1 的概率提高、被归类为 C2 的概率显著降低相关。然而，总体而言，这些模型的统计显著性不足，因此无法得出确切结论。值得注意的例外是 T0P1 和 T0P2，其估计系数符合预期，且具有很高的统计显著性。

一种可能的解释是样本数较少，使估计结果更容易受到异常值的影响。例如，T2P1 只有 299 个案例，分为 3 组并分布在 31 个分箱中。这样一来，每组平均每个分箱只有 3.25 个元素。由于观测值在整个取值范围内并非均匀分布，因此可能会出现明显的暗点，尤其是在尾部。

G. 没有多数的多数

该算法根据 Softmax 函数将观测值划分为不同组。每个观测值都被归入满足 $j^* \in \{1, \ldots, J\} = \arg\max \Pr(x_i \in X_k)$ for $k = (1, \ldots, J)$ 的组。对于类别数 $J \geq 2$，表现最佳类别的必要条件是 $\Pr(j^*|x) \geq \lim_{\delta \to 0} \left(\frac{1}{J} + \delta\right)$，其中 J 等于类别数。在我们的三类别案例中，导致分类的最低充分多数为 $0.3377 \approx \frac{1}{3} + \delta$，对应一个产量为 0.13% 的 C0 真正例。

假设分类 x_i 为 j^* 时，要求 $\Pr(j^*|x_i) \geq \alpha$，其中 $\alpha \in (0,1)$。$\alpha > \frac{1}{J}$ 的任何取值都会使分类要求更加严格。

表 VIII 展示了 $\alpha = 0.95$ 的结果。这项改动显著提升了性能，准确率达到 88.7%。对于唯一会触发交易的类别 C1，其精确率也有所提高，达到 0.933，且仅有一个被错误分类的元素来自 C0。[4] 召回率也提升至 0.875。α 的改动使错误预测仅限于相邻类别。

为了了解算法的性能如何随 α 变化，我们绘制了图表（图 13），展示给定 α 时的正确预测比例。曲线严格递增，表明在其他条件不变的情况下，提高所需分数会增加正确预测比例。

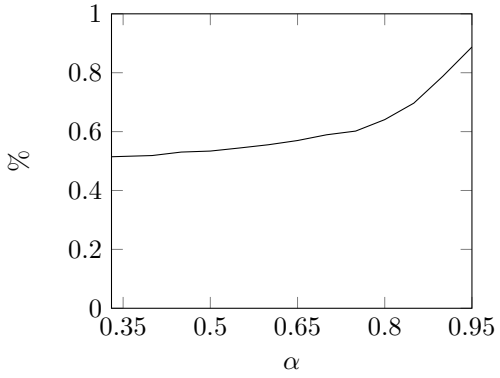

图 13. 每 α 的正确预测数

不过，这些提升也付出了代价。最明显的是，被分类的观测值数量大幅下降。例如，对于 $\alpha = 0.95$，分类数量降至整个数据集的约 2.2%（图 14）。因此，C1 预测的数量也随之减少（图 15）。

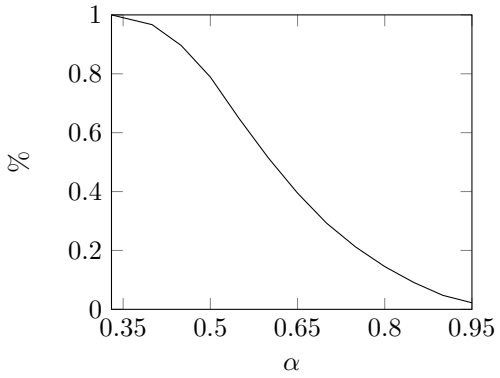

图 14. 每 α 的全部预测

两张图中的曲线在 α 中都严格递减，证明了 α 与预测数量之间的负相关关系。根据这两张图可以得出，提高要求会以数量换取质量。

为推导收益最大化水平 α，假设分类问题可以简化为两种状态 $P \in \{0, 1\}$，其中 $\gamma \in \{\gamma_0, \gamma_1\}$ 表示这两种状态各自的平均收益，$\gamma_1 \gg \gamma_0$ 和 $\phi =$ 表示 C1 预测在全部预测中的比例，取值范围为 (0,1)。那么，全部预测可以写为 $P = \phi \cdot f(\alpha) + (1 - \phi) \cdot f(\alpha)$。

[3] 在表格中，T 表示真实类别，P 表示预测类别。因此，含义隐晦的单元格 T0P1 表示：真实类别为 C0，却被错误地预测为 C1。

[4] 此外，这个元素的产量也为正，不过仅为 0.1%。

表 VII 类别级分析

	P0	P1	P2
T0	$y = \underset{(0.7476)}{0.4079} + \underset{(0.9403)}{0.0948}\,x$ $R^2 = 0.0002$	$y = \underset{(0.0095)}{-2.4136} + \underset{(0.0036)}{2.7337}\,x$ $R^2 = 0.2289$	$y = \underset{(0.0383)}{3.0056} - \underset{(0.0511)}{2.8286}\,x$ $R^2 = 0.1766$
T1	$y = \underset{(0.4453)}{2.2092} - \underset{(0.5180)}{1.8131}\,x$ $R^2 = 0.0170$	$y = \underset{(0.2256)}{-3.7526} + \underset{(0.1663)}{4.1783}\,x$ $R^2 = 0.0656$	$y = \underset{(0.0820)}{2.5434} - \underset{(0.0941)}{2.3652}\,x$ $R^2 = 0.0675$
T2	$y = \underset{(0.7459)}{-1.5705} + \underset{(0.6748)}{2.0985}\,x$ $R^2 = 0.0052$	$y = \underset{(0.7348)}{1.4877} - \underset{(0.7666)}{1.3420}\,x$ $R^2 = 0.0035$	$y = \underset{(0.8056)}{1.0828} - \underset{(0.8676)}{0.7566}\,x$ $R^2 = 0.0009$

表 VIII 95%支持率

真实值 / 预测值	0	1	2	SUM
0	46	1	5	52
1	2	14	0	16
2	0	0	3	3
SUM	48	15	8	71

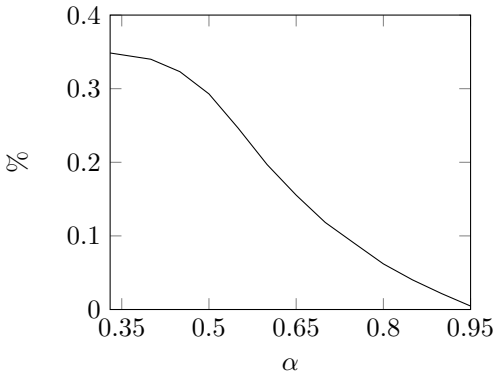

图 15. C1 预测每 α

由于只有 C1 预测会触发活动，因此可以省略右侧的第二部分，因为其期望值为 0。此外，假设 $f(\alpha)$ 是连续、可微、非负且严格单调递减的函数，用于表示预测数量与 α 之间的关系；并且 $g(\alpha)$ 是单调递增、连续、非负且可微的函数，用于表示正确预测比例与 $\alpha \in \mathcal{R} = [0,1]$之间的关系。那么，期望值可以写为 $E(Y) = \phi \cdot f(\alpha)(g(\alpha) \cdot \gamma_1 + (1 - g(\alpha)) \cdot \gamma_0)$，由此可得

$$\frac{\partial E(Y,\alpha)}{\partial \alpha} = (\gamma_1 - \mu_0)(f'(\alpha) \cdot g(\alpha) + f(\alpha) \cdot g'(\alpha)) + \gamma_0 f'(\alpha) = 0$$

为简化起见，假设 $\gamma_0 = 0$，则方程可简化为 $\frac{\partial E(Y,\alpha|\gamma_0=0)}{\partial \alpha} = \gamma_1(f'(\alpha) \cdot g(\alpha) + f(\alpha) \cdot g'(\alpha)) = 0$。我们知道 $f'(\alpha) < 0$且 $g'(\alpha) > 0$，而 $f(\alpha) \geq 0$且 $g(\alpha) \geq 0$。令 $\gamma_0 = 0$，可得 $\gamma_1 \gg 0$。为求解该方程，我们令 $f'(\alpha) \cdot g(\alpha) + f(\alpha) \cdot g'(\alpha) = 0$。由于第一个乘积非正，第二个乘积非负，因此该方程存在理论解。

一般来说，$\frac{y'(x)}{y(x)} \approx \lim_{\Delta x \to 0} \{\frac{\%\Delta f}{\%\Delta x} \cdot \frac{1}{x}\} \approx \%\lim_{\Delta x \to 0} \{\frac{\%\Delta f}{\Delta x}\}$成立，前提是 $\Delta x = x \cdot \%\Delta x$。

在我们的案例中 $f'(\alpha) \cdot g(\alpha) + f(\alpha) \cdot g'(\alpha) = \frac{f'(\alpha)}{f(\alpha)} + \frac{g'(\alpha)}{g(\alpha)} = 0$。该条件在 $\alpha^* = \alpha \in [0,1] \iff \lim_{\Delta\alpha \to 0} \frac{\%\Delta f(\alpha) + \%\Delta g(\alpha)}{\Delta\alpha} = 0$ 时满足。所写内容并不能保证存在唯一解。

V. 使用算法进行交易

为测试算法的运行情况，我们进行了两项基于仿真的实验，以模拟实时交易。两项实验采用相同的设置。首先，我们收集了股票在交易开始后一小时内的股价变动信息。其次，使用训练好的 MNET-2 算法处理这些数据。第三，根据算法的预测结果作出交易决策。未利用的机会到期并失去价值。第四，任何已建立的头寸都会在交易收盘时，或在与相应资产进行交易的最后可能时点平仓。为简化计算，假设资产是无限可分的。

A. 一次一项投资

在第一个示例中，每次只允许进行一项投资。投资机会按顺序逐一出现，并执行所有预测买入。

令 $X^P = \{x_i(p_i^0, E(p_i^1)) | \frac{E(p_i^1)}{p_i^0} \geq 1.02\}$对于 $i = 1, 2, ..., N$在开盘时股票价格为 p_i^0、i，相应的收盘价为 p_i^1，构成所有预测交易的集合，其中 $E(p)$ 表示资产 i的预期价格变动。令初始投资等于 v_0，单一头寸的最大投资额限制为 β，并令 $\{x^P\}_{t=1}^T$ 为所有预测交易组成的序列。则投资流程可以写成

$$v = \sum_{t=1}^{T} v_{t-1} + \frac{p(x_t^P)^1 - p(x_t^P)^0}{p(x_t^P)^0} \times \max(0, \min(v_{t-1}, \beta))$$

其中 p_t^0，p_t^1表示在第 $t = 1$、2、.....、T 次迭代时头寸的开盘价和收盘价。

B. 同时进行多项投资

将每个时间间隔的投资数量限制为一项，实际意义不大。在实践中，每个时间段（例如本例中的一天）都存在多个交易机会，并且它们会争夺金融资源。

表 IX 仿真结果：交易

Type	每次一项投资			每次多项投资		
	金额（美元）	交易次数	每笔交易金额（美元）	金额（美元）	交易次数	每笔交易金额（美元）
All	10,465.88	3,179	3.29	58,831.79	3,179	18.51
预测的 C1	11,397.44	1,108	10.29	73,917.22	1,108	66.71
真实的 C1	19,880.95	566	35.13	278,546.66	566	492.13
随机：50%	5,710.38	1,583	3.61	57,961.38	1,565	37.04
随机：50%	4,802.04	1,608	2.99	58,443.35	1,565	37.34
随机：33%	3,695.26	1,028	3.59	58,589.26	1,019	57.50
随机：33%	4,238.29	1,058	4.01	56,755.77	1,019	55.70
随机：33%	4,008.80	1,007	3.98	56,240.11	1,019	55.19

设 $X^P = \{x_{i,t}\ p_{i\ t}^0\ E\ p_{i\ t}^1 \quad \overline{p_{i\ t}^0} \geq 1.02\} \qquad i = (,\ ,\ (,\ ))$ $1, 2, \ldots N \qquad t = 1\ \ldots\ T$ $|E(p_{i,t}^1)$ ，其中 、 、 、和、 、构成每个时间 t 的全部预测集合，并设在 t 进行多项投资，每项投资的金额均为 s ，该金额取决于时间 v_t的累计金额和已发起的投资数量。鉴于我们无法对产量进行排序，这是一个合理的假设。进一步假设，单个资产可投资的金额存在上限 β 。通常设定该上限是为了控制风险。因此，时间 t 对某项资产的投资金额可计算为 $s_t(v_t,\ |x_{\cdot,\ t}||\beta)$ $=$ 最小值 $\left(\beta, \frac{v_t}{|x_t|}\right)$。由于不允许卖空， s_t 应严格为非负数 $s_t = \begin{cases} s_t, & \text{若 } s_t > 0, \\ 0, & \text{否则。于是，交易流程可以写为} \end{cases}$

$$v = \sum_{t=1}^{T} \left[v_{t-1} + \sum_{j=1}^{J} \frac{p(x_{j,t}^P)^1 - p(x_{j,t}^P)^0}{p(x_{j,t}^P)^0} \cdot s_t \right].$$

C. 模拟与结果

我们针对每种架构分别进行了一组实验。第一个示例的初始头寸为 $v_0^1 = 1000$美元，而第二个示例的初始投资增加至 v_0^2 =50，000美元。模拟基于2022年1月14日至2022年7月1日期间获取的真实数据进行。数据集总共包含3179个观测值，这些观测值在114个交易日内实现。

对于第一种架构，投资机会依次到来，其顺序可能会因资源积累而影响结果。为检验这一影响，我们以随机打乱的顺序进行了多次重复模拟。然而，结果表明该影响微乎其微。第二次模拟的时间单位为（交易）日，交易机会按照真实交易日的实际情况到来。在这种配置下，机会的顺序无关紧要，因为一天内所有计划进行的交易都会同时开仓和平仓。结果见表 IX。

“全部”表示启动所有可能交易的情形。“预测的 C1”表示启动算法预测为 C1 的交易的情形。“真实的 C1”表示启动收益至少为 2% 的交易的情形。在其余 5 次模拟运行中，是否交易某项资产由随机方式决定，各笔交易的概率如表所示。

在分析期间使用算法进行交易取得了正收益。为防止系统崩溃，第一种情景中的每笔投资都设有 {v1}美元的上限。在第一种情景中，使用算法进行交易，将初始的 1000 美元增至近 11,400 美元，共进行了 1108 次交易，平均每笔交易收益为 0.938%。这接近模拟得出的 0.95% 平均收益。除理论最大值外，该算法在绝对量和相对量方面的表现均优于所有其他模拟示例。例如，若所有交易都做多，平均每笔交易收益为 0.3%；而若对真实的 C1 做多，平均每笔交易收益则为 3.34%，显然是表现最佳的配置。

在第二种情景中，初始投资为 50,000 美元，算法实现了 47.8% 的总收益率。同样，它执行了 1108 笔交易，但这次是在 114 个交易日内完成的，平均每个交易日 9.7 笔交易。初始投资的日均收益为 0.343%。如果所有交易都做多，总收益率“仅”为 17.7%，即日均 0.143%，显然逊于该算法。

使用算法进行交易显著减少了交易次数。总体而言，只有 34.85% 的潜在交易实际执行了。交易次数减少带来了诸多实际益处：(1) 缩短执行时间；(2) 在资本固定的情况下，允许每笔交易投入更多资金；(3) 对于给定的投资金额，所需资本大幅减少；(4) 能够专注于高收益机会；(5) 使投资组合规模更小、更易管理。

首先，显而易见，在其他条件不变的情况下，对于固定数量的交易，一次完成的交易次数越多，执行时间就越短。在我们的案例中，如果每天只进行一笔交易，则需要 1108 天；如果允许一次进行多笔交易，则只需 114 天。若按照算法进行投资，则需要 1108 天；若所有头寸均做多，则需要 3179 天。其次，如果投资者用于投资的资金总额固定（通常确实如此），那么交易次数越少，每笔交易的投资额按定义就越大。鉴于投资中，交易次数与每笔交易的投资额之间并不具备交换性，因此，在收益率较高的较少资产上投入更多资金，可能带来更高的总回报。第三，如果要求每个头寸的规模固定，那么投资于较少数量的资产，按定义就意味着建立所有头寸所需的资金更少。这是一个理想的特点，因为资金是有成本的。第四，减少投资次数后，可以放弃收益率较低（但仍为正）的机会，将更多资金投入高收益资产，从而提高回报。事实上，这正是本文的核心。第五，较小的投资

组合更易于管理。不过，技术创新可能会减弱这种影响。

VI. 结论

本文将资产交易视为图像分类问题，并使用基于 CNN 的 MobileNet-V2 神经网络进行处理。根据价格变动，项目被分为三类：价格明显下跌、价格明显上涨，或波动过大而无法判断。我们的目标并非对价格变动幅度进行建模。模型使用纳斯达克上市公司中市值最大的股票进行训练和测试，所用数据为按 5 分钟时间间隔采集的日内数据。

在测试期间，使用该算法进行交易的表现优于市场，也优于测试中的所有交易配置，理论最大值除外。该算法能够很好地区分两个相反类别，并将大多数错误分类归入第三类——收益中性类别。使用该算法将交易次数显著减少至原来的约三分之一。所有测试均基于样本内数据集进行。

训练期间，验证阶段表现最佳的模型在三个规模平衡的类别上取得了 49.36% 的准确率。测试期间表现最佳的模型准确率为 51.5%。我们证明，如果采用更严格的分类条件，得分可以显著提高。然而，这也导致分类数量大幅减少。这显然是在质量与数量之间进行权衡。

本文的主要结果是，交易的第一个小时与收盘价之间存在一种高层次的关系，而基于 CNN 的 MobileNet-V2 模型成功提取了这种关系。鉴于这些结果，可以合理地得出结论：只要数据集的分布尽可能接近算法预期进行预测的数据，该算法就是有用的。

参考文献

[1] S. Subin 和 C. Reinicke，《美联储再次大幅加息后，道指收跌 500 点》，CNBC，9 月。 20, 2022 [ ] [ ] 2022 09 20 在线。在线。可访问：https://www.cnbc.com/ / / / stock-market-futures-open-to-close-newshtml.html [2] Z. Bodie，《普通股作为对冲通货膨胀的工具》，《金融学杂志》，第 31 卷，第 2 期，第 459–470 页，1976 年。[3] E. F. Fama 和 G. W. Schwert，《资产收益与通货膨胀》，《金融经济学杂志》，第 5 卷，第 2 期，第 115–146 页，1977 年。[4] J. Y. Campbell，《股票收益与期限结构》，《金融经济学杂志》，第 18 卷，第 2 期，第 373–399 页，1987 年。[5] D. B. Keim 和 R. F. Stambaugh，《预测股票和债券市场的收益》，《金融经济学杂志》，第 17 卷，第 2 期，第 357–390 页，1986 年。[6] K. R. French、G. W. Schwert 和 R. F. Stambaugh，《预期股票收益与波动率》，《金融经济学杂志》，第 19 卷，第 1 期，第 3–29 页，1987 年。[7] N. Barberis，《在收益可预测时进行长期投资》，《金融学杂志》，第 55 卷，第 1 期，第 225–264 页，2000 年。[8] R. F. Stambaugh，《预测回归》，《金融经济学杂志》，第 54 卷，第 3 期，第 375–421 页，1999 年。[9] I. Welch 和 A. Goyal，《全面考察股权溢价预测的实证表现》，《金融研究评论》，第 21 卷，第 4 期，第 1455–1508 页，2008 年。

[10] N. Barberis 和 R. Thaler，《行为金融学综述》，《金融经济学手册》，第 1 卷，第 1053–1128 页，2003 年。[11] J. Eapen、D. Bein 和 A. Verma，《结合卷积神经网络和双向长短期记忆网络的全新深度学习模型，以改进股票市场指数预测》，载于 2019年IEEE第九届计算与通信研讨会及会议（CCWC）。IEEE，2019 年，第 0264–0270 页。[12] A. Tsantekidis、N. Passalis、A. Tefas、J. Kanniainen、M. Gabbouj 和 A. Iosifidis，《利用卷积神经网络根据限价订单簿预测股票价格》，载于 2017年IEEE第十九届商业信息学会议（CBI），第 1 卷。IEEE，2017 年，第 7–12 页。[13] H. S. Sim、H. I. Kim 和 J. J. Ahn，《用于图像识别的深度学习是否适用于股票市场预测？》，Complexity，第 2019 卷，文章编号 4324878，2019 年。 [在线]。可访问：https://doi.org/ 10 1155 2019 4324878 。/ / [14] M. U. Gudelek 、 S. A. Boluk 和 A. M. Ozbayoglu，《基于深度学习的股票交易模型：利用二维卷积神经网络进行趋势检测》，载于 2017年IEEE计算智能专题系列会议（SSCI）。IEEE，2017 年，第 1–8 页。[15] O. B. Sezer 和 A. M. Ozbayoglu，《利用深度卷积神经网络进行算法金融交易：时间序列到图像转换方法》，Applied Soft Computing，第 70 卷，第 525–538 页，2018 年。[16] W. Long、Z. Lu 和 L. Cui，《基于深度学习的股票价格变动预测特征工程》，Knowledge-Based Systems，第 164 卷，第 163–173 页，2019 年。[17] W. Chen、M. Jiang、W.-G. Zhang 和 Z. Chen，《一种用于股票趋势预测的新型基于图卷积特征的卷积神经网络》，Information Sciences，第 556 卷，第 67–94 页，2021 年。[18] Z. Hu、Y. Zhao 和 M. Khushi，《利用深度学习进行外汇与股票价格预测综述》，Applied System Innovation，第 4 卷，第 1 期，文章编号 9，2021 年。[19] O. B. Sezer、M. U. Gudelek 和 A. M. Ozbayoglu，《基于深度学习的金融时间序列预测：系统性文献综述：2005–2019》，Applied Soft Computing，第 90 卷，文章编号 106181，2020 年。[20] M. Sandler、A. Howard、M. Zhu、A. Zhmoginov 和 L.-C. Chen，《MobileNetV2：倒置残差与线性瓶颈》，载于 IEEE计算机视觉与模式识别会议论文集，2018 年，第 4510–4520 页。[21] I. Goodfellow、Y. Bengio 和 A. Courville，《深度学习》。麻省理工学院出版社，2016 年。[22] R. Aroussi，yfinance，版本 0.1。85，2022 [在线]。 [在线]。可访问：https://github.com/ranaroussi/yfinance.git [23] D. Goldfarb，mplfinance，版本 0.12.9b5，2022 [在线]。 [在线]。可访问：https://pypi.org/project/mplfinance/ [24] L. Rice、E. Wong 和 Z. Kolter，《对抗鲁棒深度学习中的过拟合》，载于第 37 届国际机器学习会议论文集，《机器学习研究论文集》系列，H. D. III 和 A. Singh 编，第 119 卷。PMLR，2020 年 7 月 13–18 日，第 8093–8104 页。 [在线]。可访问：https://proceedings.mlr.press/v119/rice20a.html [25]N. Srivastava、G. Hinton、A. Krizhevsky、I. Sutskever 和 R. Salakhutdinov，《随机失活：一种防止神经网络过拟合的简单方法》，机器学习研究期刊，第 15 卷，第 56 期，第 1929–1958 页，2014 年。 [在线]。可访问：http://jmlr.org/papers/v15/srivastava14a.html [26] D. Warde-Farley、I. J. Goodfellow、A. Courville 和 Y. Bengio，《分段线性网络中随机失活的实证分析》，2013 年。[在线]。可访问：https://arxiv.org/abs/1312.6197



Matej Steinbacher 毕业于斯洛文尼亚的马里博尔大学经济学专业，随后在斯洛文尼亚的卢布尔雅那大学经济学院继续攻读研究生。他的研究重点是经济学和金融领域的网络模型。